IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 22, 2019, accepted December 13, 2019, date of publication December 17, 2019,
date of current version December 31, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2960263*

# Hierarchical Data Augmentation and the Application in Text Classification

**SHUJUAN YU[1], JIE YANG[1], DANLEI LIU[1], RUNQI LI[1], YUN ZHANG[1], AND SHENGMEI ZHAO[2]**

[1]College of Electronic and Optical Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[2]College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Corresponding author: Jie Yang (1017020718@njupt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61871234.

**ABSTRACT** The applications of data augmentation in natural language processing have been limited. In this paper, we propose a novel method named Hierarchical Data Augmentation (HDA) which applied for text classification. Firstly, inspired by the hierarchical structure of texts, as words form a sentence and sentences form a document, HDA implements a hierarchical data augmentation strategy by augmenting texts at word-level and sentence level respectively. Secondly, inspired by the cropping, a popular method of data augmentation in computer vision, at each augmenting level, HDA utilizes attention mechanism to distill (crop) important contents from texts hierarchically as summaries of texts. Specifically, we use a trained Hierarchical Attention Networks (HAN) model to obtain attention values of all documents in training sets at both levels respectively, which are further used to extract the most important part of words/sentences and generate new samples by concatenating them in order. Then we gain two levels of augmented datasets, WordSet and SentSet. Finally, extending training set with certain amount of HDA-generated samples and we evaluate models' performance with new training set. The results reveal HDA can generate massive and high-quality augmented samples at both levels, and models using these samples can obtain significant improvements. Compared with the existing methods, HDA enjoys the simplicity both on theory and implementation, and it can augment texts at two levels for the diversity of data.

**INDEX TERMS** Attention mechanism, data augmentation, natural language processing, text classification.

## I. INTRODUCTION

Recently, with the prosperity of deep learning, tremendous successes have been witnessed in many areas such as computer vision [1], speech recognition [2], and natural language processing (NLP) [3]. Typically, the quantity and quality of training data are of great significance to the generalization performance of models in deep learning [4]. Overfitting and high generalization error are prone to arise under insufficient training data [5], [6]. However, preparing a large-scale labeled dataset is time-consuming and laborious [7], [8]. Consequently, there is a surge of research in data augmentation.

Data augmentation aims at creating additional data by generating variants of existing data through transformations [4], [10]. It has got successful applications in computer vision such as deep generative model [11], Variational Autoencoders [12], GANs [13]. Nevertheless, data augmentation is widely acknowledged to be difficultly applied to natural language processing [14]. A main reason comes from the discrete nature of text due to the limited size of vocabulary of a language, the resulting non-differentiability hinders the applications of plenty of technologies proven to be very effective in computer vision [15]; Another important reason is that the poor anti-interference ability of texts, as small perturbations on texts may completely change the meaning [6], [10]. E.g., if we delete word *never* from *I never like apple*, the sentiment of this sentence will be completely changed (from negative to positive). Currently, although there are various kinds of techniques proposed for the applications of data augmentation in NLP, universal methods have not been explored [16]. Traditional methods include synonym replacement (SR) [6], [17], Back-translation [18], and task-specific heuristic rules [19].

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

In view of this, we firstly propose a simple but powerful data augmentation technique called Hierarchical Data Augmentation (HDA), which features with augmenting texts via distilling their essential contents at sentence-level and word-level with attention mechanism [21], [22] respectively. The motivation of our work originates from the following two aspects: Firstly, texts naturally has a hierarchical structure [20], as words form a sentence and sentences form a document, thus it is reasonable to augment texts at word level and sentence-level respectively for the diversity of data; Secondly, inspired by the cropping [25], a popular data augmentation method for Computer Version which generates new samples by randomly cropping the informative part of content from the original images [26], at each augmenting level, HDA generates new samples by extracting (cropping) a part of important contents from texts with attention mechanism. Considering that either for images or texts [20], [22], different parts are different important to the whole, and attention mechanism can effectively measure the importance of parts with regard to the whole. Thus we can use it to extract the important words/sentences from texts to generate new samples. And since HDA-generated samples keep the most important part of the original samples, thus can preserve the labels unchanged.

Generally, HDA consists of two levels of hierarchical attention-based augmentation components, sentence-level attention augmentation (SentAtt) and word-level attention augmentation (WordAtt). Specifically, on the basis of the work of Hierarchical Attention Networks (HAN) [20], we firstly train a HAN model with training set, and use its sentence-level attention mechanism to obtain the sentence-level attention values for each sample in the training set. Then we extract the top $T_s$ percent of the most important sentences from samples according to attention values, and generate a new sample by concatenating these sentences in order. And performing SentAtt over the whole training set, a sentence-level augmented dataset, SentSet can be obtained. Analogously, we use HAN's word-level attention mechanism to obtain the word attention values for every sentence in all samples. Then we randomly delete 50% of words belonging to the top $T_w$ percent of the most unimportant of words for long sentences in a sample and keep the rest sentences unchanged, in this way we obtain an augmented sample. Via performing WordAtt over the whole training set, another new augmented dataset named WordSet can be produced as well. Moreover, in order to make full use of the hierarchical information of HAN, a hierarchical dataset named HybridSet is further generated by randomly extracting equal number of samples from SentSet and WordSet. Finally, we extend original training set with certain numbers of samples from WordSet, SentSet, and HybirdSet respectively, and we evaluate the performance of models with HDA-extended training set.

In this paper, we apply HDA for long-text classification task. In order to verify the generality of HDA, we choose two different HAN models as baseline models, i.e., vanilla HAN and its variant, Hierarchical Attentional Hybrid Neural Networks (HAHNN) [23], and we evaluate models on two long-text datasets, IMDb and Yelp. The experimental results reveal that HDA is a promising technique as it 1) Compared with Easy Data Augmentation (EDA), HDA can not only augment texts much more faster, but also generate massive augmented samples; 2) HDA-generated samples are proven to be high-quality for training since the more samples added to the training set, the better models can perform.

To summarize, the contributions of our work lie in the following three points:

- Taking the hierarchical structure of texts into account, the proposed HDA firstly augments texts at word-level and sentence-level respectively, and previous work is mainly at single level.
- Based on attention mechanism, at each augmenting level, HDA generates new samples by extracting the most important part of the contents from texts, which can keep the labels unchanged.
- The results of our experiments denote that EDA degrades the performances of models on long-text datasets, which proves that EDA is not a satisfying data augmentation for text classification.

## II. RELATED WORK

Recently, a variety of data augmentation techniques have been explored for NLP. And synonym replacement (SR) [6], [17] may be the one of the most simple and intuitive approaches among them to augment texts by randomly replacing words with one of their synonyms. However, Due to the number of synonyms of words is very limited, SR can not produce various data, and needs extra knowledge of a language, i.e., thesaurus. Different from [6], [17], Contextual Augmentation [5] (CA) replaces words with other words predicted by a language model (LM) according to the contexts, thus can offer a wider range of substitute words than SR. However, for implementing CA, one should first pre-train a LM on a very large corpus without label-conditional architecture, next the LM should be further trained with label-conditional architectures to predict substitute words and augment texts. Thus CA is quite a complex method for data augmentation. Proposed in [16], Easy Data Augmentation (EDA) is an upgraded version of SR, which consists of four different sub-operations: synonym replacement, random insertion, random swap, and random deletion. Each operation randomly changes (adds, replaces, or deletes) few words in a sentence to generate a new one. And then, an augmented sample is produced by extending original sentence with four EDA-generated sentences. Obviously, EDA augments texts completely in a random way, which may lead to unreliable and uncontrolled data [15]. Besides, The EDA-augmented samples have severe information redundancy for nearly 5 times larger the amount of data compared to the original sample (shown in Fig. 2). Lastly, the above three methods can only augment texts at word-level.

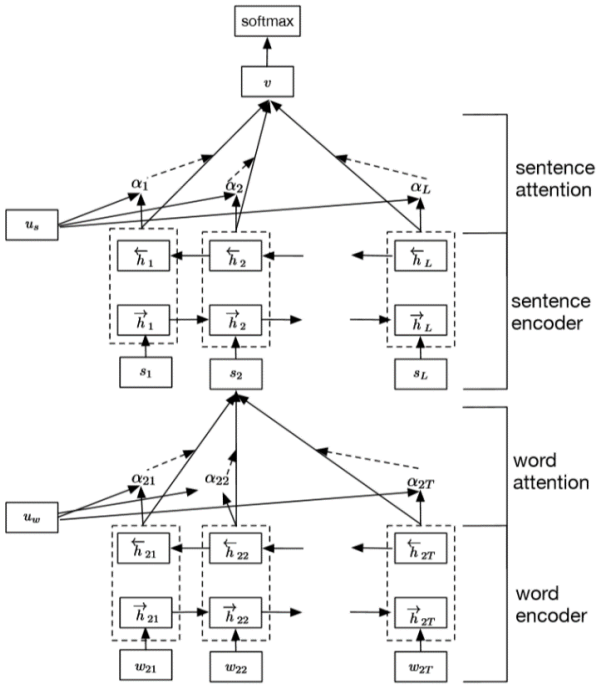



**FIGURE 1.** The architecture of HAN [20].

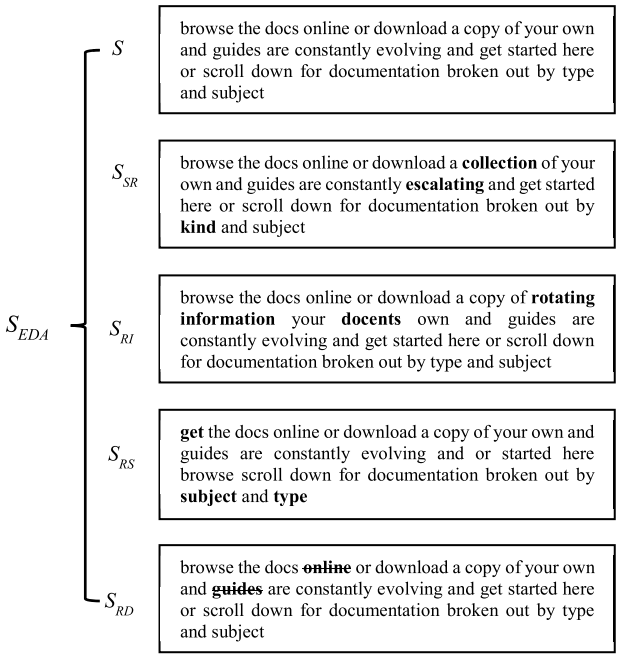



**FIGURE 2.** The visualization of EDA. Words edited by EDA are highlighted in bold. Specially, sample $S$ consists of 31 words, which means SR, RI, and RS will change 3 words exactly and RD will change 3 words averagely. Here RD deletes two words from $S$, i.e., ***online*** and ***guides***.

Featuring with augmenting the parallel training corpus with back-translations of target language sentences [24], Back-translation [18] is another popular technique which can generate diverse phrases with keeping the semantics of the texts unchanged and thus shows great improvements for machine translation and question answering [4], [16]. Nevertheless, Back-translation [18] is a very task-specific technique and can not be applied to other tasks in NLP like text classification.

Based on Variational Autoencoders and holistic discriminators, [15] proposed a text generative model named Vae+Dis to generate realistic sentences with controlled sentiments and tenses, whose attributes are controlled by learning disentangled latent representations with designated semantics. The results denote that Vae+Dis can outperform over previous generative models on the accuracy of generating specified attributes as well as performing classification tasks. However, this method can only augment texts at sentence-level as well.

Different from all above methods, the most exciting advantage of our method is that with its two levels of hierarchical augmentation, HDA can augment texts both at word-level and sentence-level, thus HDA can generate massive and diverse data. Besides, based on the existing HAN models, HDA only introduces simple operations which extract the important sentences/words according to attention values, and is free of requiring any extra support like thesaurus. Moreover, HDA is very friendly for implementation, which is another significant advantage compared to above models.

## III. BACKGROUND

In this section, we will describe the work of HAN and EDA.

### A. HAN

Texts naturally have a hierarchical structure, e.g., a sentence is formed by words, and similarly, a document is formed by sentences. As shown in Figure 1, in order to capture the hierarchical information of texts, HAN designs a hierarchical structure which stacks two levels of modules to capture the information at word-level and sentence-level respectively. And each level of module consists of an encoder layer and an attention mechanism layer. Specially, we use bidirectional GRU [24] as encoder.

Given a document $D = [S_1, S_2, \ldots, S_L]$, where $L$ is the number of sentences in $D$ and $S_i (1 \leq i \leq L)$ represents the $i^{th}$ sentence in $D$. And for $S_i = [W_{i1}, W_{i2}, \ldots, W_{iT}]$, $W_{it} (1 \leq t \leq T)$ represents the $t^{th}$ word in $S_i$. The bidirectional GRU uses two GRUs to encode sentence in bidirectional directions: forward and backward. And the calculations of the forward and backward GRU can be formulated as follows:

$$\overrightarrow{h_{it}} = \overrightarrow{GRU}(W_{it}), \quad t \in [1, T] \tag{1}$$

$$\overleftarrow{h_{it}} = \overleftarrow{GRU}(W_{it}), \quad t \in [T, 1] \tag{2}$$

The representation of $W_{it}$ can be obtained by concatenating $\overrightarrow{h_{it}}$ and $\overleftarrow{h_{it}}$, i.e., $h_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$, which summarizes the information of the whole sentence centered around $W_{it}$.

Then HAN introduces word-level attention mechanism to quantify the importance of $W_{it}$ to the meaning of $S_i$ and aggregate the representation of those words to form a sentence vector $s_i$, specially:

$$u_{it} = \tanh(W_w h_{it} + b_w) \tag{3}$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{4}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{5}$$

Concretely speaking, equation (3) denotes that firstly a hyperbolic tangent activate function is used to obtain a representation of $h_{it}$, which is denoted as $u_{it}$. $W_w$ and $b_w$ are the weight matrix and bias respectively. Then HAN measures the importance of a word as the similarity of $u_{it}$ with a word level context vector $u_w$, and get a normalized importance weight $\alpha_{it}$(word-level attention value) via a softmax function (shown in equation (4)). And $\alpha_{it}$ quantifies the importance of $W_{it}$ to sentence $S_i$. After that, equation (5) computes the sentence vector $s_i$ as a weighted sum of the word representations based on the attention value $\alpha_{it}$. What's more, word context vector $u_w$ is randomly initialized and jointly learned during the training process.

Given the sentence vector $s_i$, and the forward and backward hidden states of $s_i$ can be obtained as follows:

$$\overrightarrow{h_i} = \overrightarrow{GRU}(s_i), \quad t \in [1, L] \tag{6}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(s_i), \quad t \in [L, 1] \tag{7}$$

Then the annotation of $s_i$ can be denoted as $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$, which summarizes the information of sentence $S_i$ and its neighbor sentences. Similarly, HAN introduces sentence-level attention mechanism to quantify the importance of the sentences to a document:

$$u_i = \tanh(W_s h_i + b_s) \tag{8}$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \tag{9}$$

$$v = \sum_i \alpha_i h_i \tag{10}$$

where $v$ is the document vector which summarizes all the sentences in the document $D$ and can be used as features for classification. $W_s$ and $b_s$ denote the weight matrix and bias. Equation (8) computes the representation of $h_i$, which is denoted as $u_i$, and equation (9) denotes the computation of sentence-level attention values. Equation (10) denotes the computation of $v$. Additionally, $u_s$ is a sentence-level context vector, which is also randomly initialized and jointly learned during the training processing.

Moreover, the document vector $v$ can be regarded as a high-level representation of the documents, which is rational to be used as features for classification. Hence, HAN uses a softmax layer which receives $v$ as input to predict the probability distribution for classification as follows:

$$p = \text{softmax}(W_c v + b_c) \tag{11}$$

where $p$ denotes the probability distribution predicted by HAN, $W_c$ and $b_c$ denote the weight matrix and bias.

Considering that HAHNN is a simple extension of HNN as it only introduces a CNN layer after word embedding, thus the introduction of HAHNN is omitted for simplicity.

### B. EDA

Given a sentence $S = [W_1, W_2, \ldots, W_T]$, $T$ is the length of $S$, and $W_t (1 \leq t \leq T)$ represents the $t^{th}$ word in $S$. EDA augments $S$ by performing four simple but independent sub operations as follows:

- Synonym Replacement (SR): Randomly choose $n$ words from $S$ which are not stop words, then replace each of them with any of its synonyms randomly and obtain a new sentence $S_{SR}$.
- Random Insertion (RI): Randomly choose a word from $S$, and insert any of its synonyms to any position in $S$ randomly. Repeat above operation $n$ times and obtain a new sentence $S_{RI}$.
- Random Swap (RS): Randomly choose 2 words from $S$ and swap their positions. Repeat $n$ times and obtain a new sentence $S_{RS}$.
- Random Deletion (RD): Randomly delete each word from $S$ with probability $p$ and obtain a new sentence $S_{RD}$.

Specially, in the above operations, $n = \alpha T$, where $\alpha$ is a parameter that represents the changed percent of $T$ words. And for RD, $p = \alpha$. In other words, EDA varies the number of changed words according to the length of the sentence. And for those datasets sizing more than 5000 samples, EDA recommends that $\alpha = 0.1$. Finally, $S$ is extended with four augmented sentences and have $S_{EDA} = [S, S_{SR}, S_{RI}, S_{RS}, S_{RD}]$. The visualization of EDA is shown in Figure 2.

Considering that all four sub operations only change small percent of $T$ words in $S$, thus every sentence in $S_{EDA}$ shares high similarity with each other. Furthermore, there is severe information redundancy in EDA since it concatenates 5 highly similar sentences together and the amount of data grows nearly to 5 times compared to the original (shown in Fig. 2), which makes massive pressure both on model training and data storage, especially for long-text classification task.

In terms of long-text datasets, it is unwise to perform EDA over every sentence for documents in datasets. For instance, assume that a document $D = [S^1, S^2, \ldots, S^L]$, where $L$ is the number of sentences in $D$ and $S^i$ $(1 \leq i \leq L)$ represents the $i^{th}$ sentence in $D$. If we perform EDA over $S^i$ and gain $D' = [..., S^i, S^i_{SR}, S^i_{RI}, S^i_{RS}, S^i_{RD}, ...]$ $(1 \leq i \leq L)$. And for $2 \leq i \leq L$-1, the contextual information of $S^i$ changes from $S^{i-1}$ and $S^{i+1}$ to $S^{i-1}_{RD}$ and $S^i_{SR}$. Note that $S^i_{SR}$ shares high similarity with $S^i$, thus $S^i$ is followed by itself rather than the subsequent sentence $S^{i+1}$ after EDA. Apparently, it is an unrealistic pattern which hardly appears in real texts (we hardly repeat a sentence twice in writing), whereas may be learnt by models as features and lead to overfitting. Considering that the contextual information is important to text classification, thus we can deduce that perform EDA over all the sentences in $D$ will not only destroy the contextual

information in documents but also significantly increase the amount of data (5 times larger than the original).

In view of this, we follow the recommendation of EDA and apply EDA for long-text datasets in the following way: Firstly we set $\alpha = 0.1$, and for each document $D$ in training set, we randomly choose a sentence $S^i$ $(1 \leq i \leq L)$ and perform EDA over it, finally we replace $S^i$ with $S^i_{EDA}$ in $D$ and have $D' = [S^1, S^2, \ldots, S^{i-1}, S^i, S^i_{SR}, S^i_{RI}, S^i_{RS}, S^i_{RD}, S^{i+1}, \ldots, S^L]$.

We perform EDA over chosen datasets and evaluate the performance of models with EDA-augmented training set. The results are listed in Table 3. And we can easily observe that EDA degrades the performance of models on all four experiments compared to baselines, e.g., the performance of HAN with EDA decreases by 0.12% and 0.06% on IMDb and Yelp; And for HAHNN with EDA, the performance decreases by 0.16% and 0.09%. Note that EDA extends all the documents with four augmented sentences. In other words, a larger amount of data brings worse performance, which indicates that EDA is an unsatisfying data augmentation technique as it cannot generate high-quality data for training despite adding several sentences into documents. Considering the superiority on short-text datasets, the application of EDA is very limited.

## IV. HDA

As mentioned before, for the propose of augmenting texts hierarchically, the proposed HDA has a hierarchical structure of attention-based augmentation, which consists of sentence-level attention augmentation (SentAtt) and word-level attention augmentation (WordAtt). And at each level, HDA augments texts with attention mechanism. Considering that either for texts or images, different parts are different informative, e.g., it is acknowledged that there are differentiations on words' contribution to the meaning of whole sentence. And attention mechanism can quantify (via attention values) the importance of words to a sentence, thus guide models to pay less or more attention to individual words when training. Because of its sensitivity to the information, HDA applies attention mechanism for data augmentation in text classification task, more specifically, to augment texts at word-level and sentence-level.

In order to perform HDA over datasets, we only need to train a HAN/HAHNN model with original training set, and then refeed models with original training set to obtain attention values on sentence-level and word-level for every document. Then new samples can be generated by extracting the most important part of sentences or words from documents according to attention values and concatenating them in order. The core of SentAtt and WordAtt is to highlight the most important (informative) contents of the texts by deleting the part of unimportant (uninformative) contents from texts, which is similar to noise reduction.

Specifically, assume that a document $D = [S_1, S_2, \ldots, S_L]$, $L$ is the number of sentences in $D$, and $S_i (1 \leq i \leq L)$ represents the $i^{th}$ sentence in $D$. $S_i = [W_{i1}, W_{i2}, \ldots, W_{iT}]$, where $T$ is the length of $S_i$ and $W_{it}$ $(1 \leq t \leq T)$ is the $i^{th}$ word in $S_i$. For SentAtt, we only perform it over the documents consisting of more than 10 sentences (i.e., $L \geq 10$), and directly extract the most important top $T_s$ ($T_s$ is the threshold for SentAtt) percent sentences from each of those documents as well as concatenate them in order and finally generate an augmented document. Repeat above operation through all the training set and a sentence-level augmented set, SentSet, can be obtained.

**TABLE 1. The Comparison between HDA and EDA on time complexity and augmenting effort, "s" in column time denotes second.**

| Method | | IMDb | | Yelp | |
|---|---|---|---|---|---|
| | | Time | Samples | Time | Samples |
| HDA | SentAtt | 26s | 26359 | 263s | 128103 |
| | WordAtt | 33s | 44078 | 547s | 450000 |
| EDA | | 195s | 45000 | 3507s | 450000 |

As for WordAtt, we only perform it over the documents having at least one sentence which consisting of more than 10 words (i.e., $T \geq 10$). And different from SentAtt, we do not directly extract the most important words according to the word-level attention values for those sentences. Instead, in order to add certain noise to enrich the diversity of data generated by WordAtt, taking inspiration from smoothness enforcing methods [4], we randomly delete the words belonging to the most unimportant top $T_w$ ($T_w$ is the threshold for WordAtt) words with probability 50% and keep the rest sentences unchanged, thus small and unimportant perturbations (these words are unimportant to the meaning of the sentence, which can be treated as noise) can be introduced when performing WordAtt to enrich the diversity of data. In this way we obtain a word-level augmented document. Repeat above operation through all the training set and a word-level augmented set, WordSet can be obtained. The statistics of performing SentAtt/WordAtt over chosen datasets are listed in Table 1. And the illustration of performing SentAtt, WordAtt on a document (Fig. 4 (a)) is shown in Fig. 4 (b), (c) respectively.

The above discussion denotes that HDA enjoys the simplicity both on theory and implementation. In order to have a deeper understanding of this point, we conduct a set of experiments to make a brief comparison on time complexity and augmenting effort when performing HDA and EDA on IMDb and Yelp. The results are listed in Table 1. From Table 1, we can learn that on both datasets, HDA can generate more samples with less time compared to EDA. For instance, on IMDb, HDA spends only 59s to generate 70437 samples in total, whereas EDA takes 195s to generate 45000 samples. In other words, HDA generates 56.5% more samples in merely 30% of time spent by EDA. And similar results can be observed in Yelp as well. Table 1 denotes that HDA is a simpler but more powerful method compared to EDA.

We further conduct a set of experiments to find which $T_s$ and $T_w$ are the optimal values for SentAtt and WordAtt. The statistics are denoted in Fig. 3 (a), (b). Specially, for the propose of simplicity and convenience, here this set of

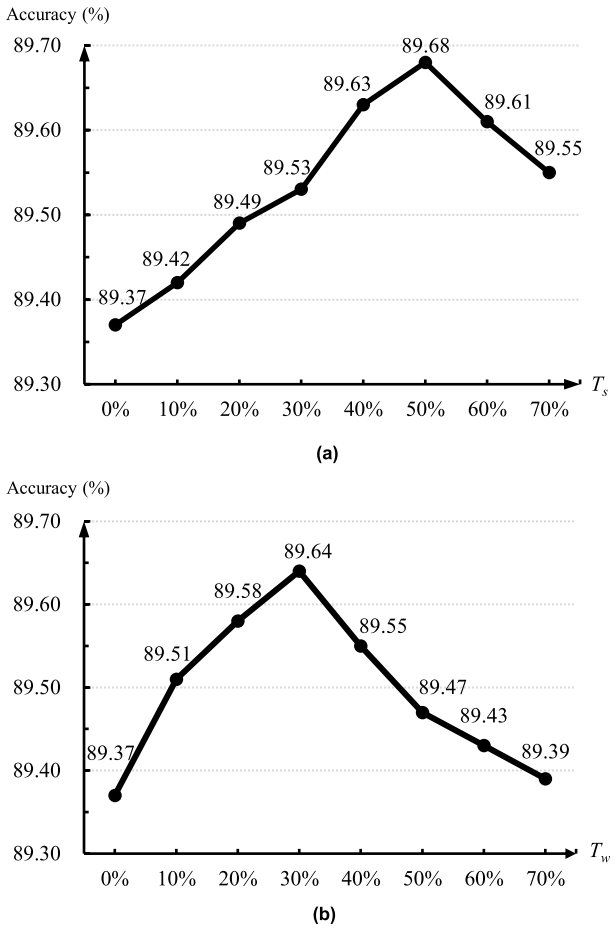



**FIGURE 3.** The accuracy curve of HAN with the increase of $T_S$ (a) and $T_W$ (b). Specially, 0% means the accuracy of vanilla HAN with original training set.

experiments are conducted only on vanilla HAN with IMDb, and we fix to extend 10% of the training set with the samples randomly extracted from SentSet or WordSet respectively. And from Fig. 3 (a), (b), we can easily find that $T_s = 50\%$ and $T_w = 30\%$ are the optimal values for SentAtt and WordAtt. Thus in all the following experiments, we set $T_s = 50\%$ and $T_w = 30\%$.

What's more, we further generate another augmented set named HybridSet to make full use of hierarchical information of texts by randomly extracting equivalent number of samples from SentSet and WordSet. For instance, if we want to extend 10% of the training set with HybridSet, we just extend 5% of the training set with SentSet and WordSet respectively.

## V. EXPERIEMNTS

We utilize two long-text datasets for experiments as follows:

- IMDb Reviews (IMDb): Obtained from [26], IMDb contains 50000 labeled movie reviews which corresponds a binary sentiment classification (positive and negative).
- Yelp 2018 (Yelp): Yelp is obtained from Yelp Dataset Challenge in 2018 which consists of more than 1.5m samples which corresponds a 5-class classification (rating from 0-5 stars). Specially, following with HAHNN, we fix in 500k the evaluated size.

**TABLE 2.** The statistics of SentSet & WordSet on IMDb and Yelp.

| | IMDb | Yelp |
|---|---|---|
| Size of training set | 45000 | 450000 |
| Size of SentSet | 26359 | 128103 |
| Size of WordSet | 44020 | 440730 |

**TABLE 3.** The comparison of performance of models with EDA and HDA, vanilla HAN and HAHNN are used as baseline models. The best and worst results in each experiment are marked in bold and italic respectively.

| Model | Yelp | IMDb | Model | Yelp | IMDb |
|---|---|---|---|---|---|
| HAHNN | 72.88% | 90.71% | HAN | 73.91% | 89.37% |
| +EDA | *72.81%* | *90.55%* | +EDA | *73.85%* | *89.25%* |
| +SentSet | 73.55% | 90.94% | +SentSet | 74.21% | **89.68%** |
| +WordSet | 73.60% | 91.02% | +WordSet | **74.31%** | 89.64% |
| +HybridSet | **73.93%** | **91.12%** | +HybridSet | 74.24% | 89.58% |

In this paper, all the experiments are performed on a computer with NVIDIA RTX 2070 for GPU acceleration, Inter (R) Core (TM) i7-8700k CPU and 16G memory. And all the models are developed under the deep learning development framework of TensorFlow.

For model details, on both datasets, we use 90% of the data for training sets and the rest 10% for testing sets. The dimension of word embedding is 200, which is obtained by the FastText [27]. The hidden size of GRU is 50. For training details, we use the Adam optimizer [29] with shuffled mini-batch. The learning rate and batch size are set to 0.001 and 128 respectively.

For preprocessing details, we limit the number of sentences in all samples to 20, for those samples consisting of more than 20 sentences, we just keep the first 20 sentences, and for those samples consisting of less than 20 sentences, we add several "empty" sentences (each consists of 20 zeros) with zero-padding [30].

Besides. We limit the number of words for all the sentences to 50. Similarly, for those sentences consisting of more than 50 words, we just keep the first 50 words, and for those sentences consisting of less than 50 words, we add several "words" with zero-padding. Specially, for EDA-generated samples, considering that EDA will add another 4 sentences to original samples, thus in the end we adjust the limitation of sentences to 24 so as to keep the original sentences information consistent with HDA.

From Table 2 we can easily learn that HDA can generate massive augmented samples on both levels of augmentation, e.g., SentAtt can produce documents equal to 58.5% of IMDb and 28.5% of Yelp, and WordAtt can produce documents equal to 97.8% of IMDb and 97.9% of Yelp. In other words, WordAtt augments nearly all the training set, whereas SentAtt only augments the longer documents in training set. We first randomly extract samples equivalent to 10% of training set from SentSet, WordSet and HybridSet respectively and extend training set with these samples. Then we evaluate the performance of models with extended training set. Results are listed in Table 3.

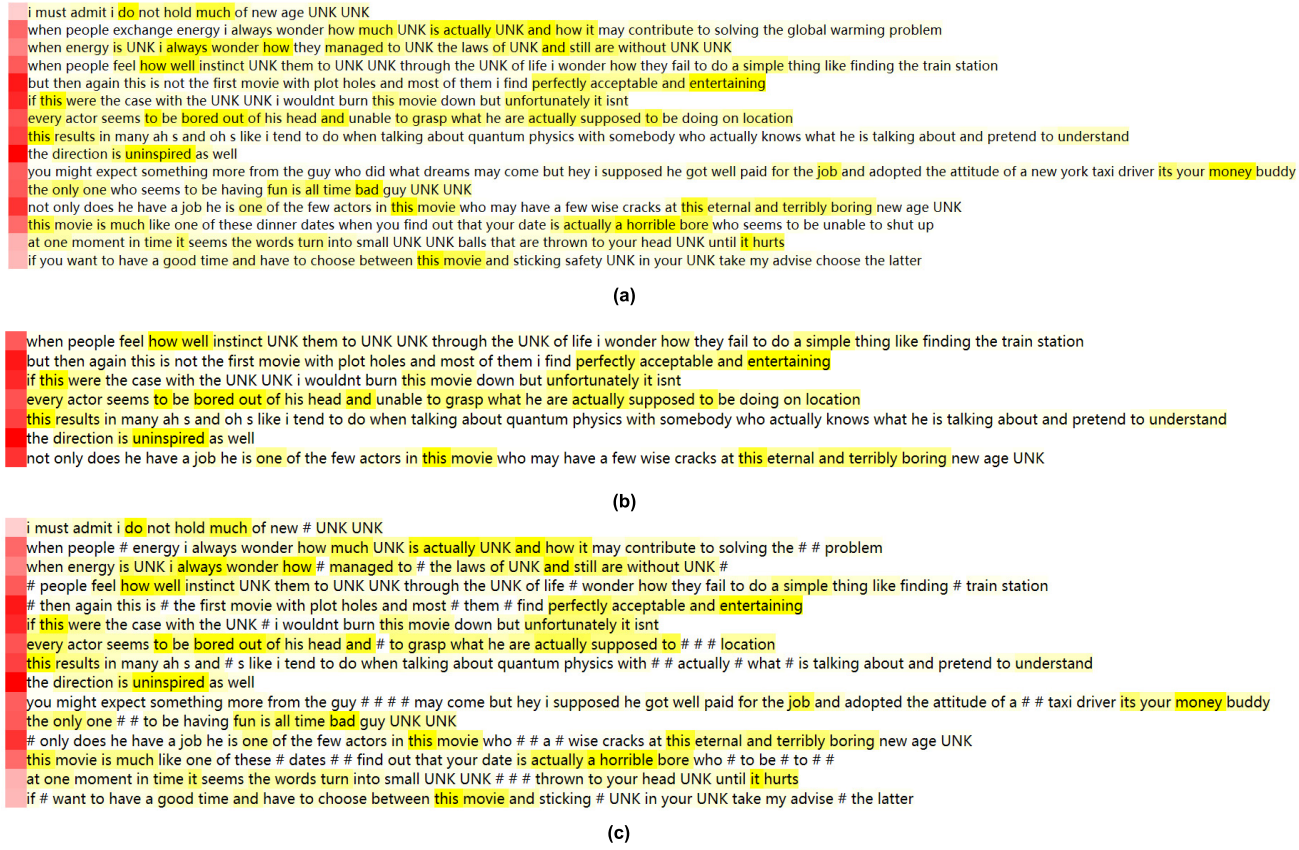



**FIGURE 4.** The illustration of a document (a) and its corresponding SentSet (b) and WordSet (c). The depth of red and yellow denote the importance of words to sentence and sentences to document respectively. Specially, here $T_S$ and $T_W$ are set to 50% and 30% respectively, and in order to better display the work of WordAtt, we explicitly use token "#" to mark the words being deleted in (c), and token "#" does not appear in WordSet actually.

From Table 3, we can learn that despite extending only 10 % of original training set with the samples from SentSet, WordSet, and HybridSet, models can continuously show significant superiority over baselines with all the four experiments, i.e., HAN with SentSet, WordSet, and HybridSet obtain a gain of 0.31%,0.27%, and 0.21% on IMDb respectively. As for HAHNN, gain is 0.23%,0.31% and 0.41%. Moreover, on Yelp, HAN with SentSet, WordSet, and HybridSet obtain a gain of 0.30%,0.40%, and 0.33% respectively. As for HAHNN, gain is 0.67%,0.72%, and 1.05% respectively.

As a result, we can conclude that firstly, HDA is a promising data augmentation technique as it can convincingly outperform baselines across all the experiments. With extending samples equals to 10% of the training set from SentSet, WordSet, and HybridSet, models can perform significantly better than baselines; However, EDA is an unsatisfying technique for its continuous degradation on performance in all the four experiments, which denotes that the EDA-generated data have adverse effect for training. What's more, although models perform better than baselines with the help of SentSet, WordSet, and HybridSet, we just cannot know the difference among SentSet, WordSet, and HybridSet from Table 3. I.e., among four experiments, SentSet and WordSet win the first place once respectively, and HybridSet wins the first place twice. Thus, it is hard to deduce that which augmented dataset is the best based on the current statistics.

Considering that we haven't made full use of all the data produced by SentAtt and WordAtt at present, it is worthwhile to evaluate the performance of models with the increasing number of samples from SentSet, WordSet, and HybridSet added to training set. Hence, we further conduct a set of experiments by extending the percentage of training set with new samples from 0% to 50% with 5% as interval. Obviously, 0% means using original training set to train models without any help of HDA. The results are listed in Table 4. Considering that HAHNN is a simple variant of vanilla HAN, thus this time we only perform experiments on vanilla HAN for simplicity and convenience.

From Table 4, it is easy to observe that the size of datasets is crucial to the performance of the model. With the increase number of samples added to training set, HAN with SentSet, WordSet, and HybridSet perform continuously better and better. Specially, when at maximum percent, HAN obtain its optimal performance on all the six experiments. For instance, at 50%, HAN with SentSet, WordSet, and HybridSet can obtain another gain of 0.61%, 0.50%, and 0.9% compared to the performance at 10% on IMDb. As for Yelp, HAN can

**TABLE 4. The Performance of model with the increasing amount of SentSet, WordSet, and HybridSet added to training set, evaluated only on vanilla HAN. The column Percent represents that the percentage of extending samples in original training datasets.**

| | IMDb | | | | Yelp | | |
|---|---|---|---|---|---|---|---|
| Percent | +SentSet | +WordSet | +HybridSet | Percent | +SentSet | +WordSet | +HybridSet |
| 0% | | 89.37% | | 0% | | 73.91% | |
| 5% | 89.60% | 89.64% | 89.45% | 5% | 74.11% | 74.14% | 74.10% |
| 10% | 89.68% | 89.64% | 89.58% | 10% | 74.21% | 74.31% | 74.24% |
| 15% | 89.76% | 89.70% | 89.70% | 15% | 74.33% | 74.34% | 74.32% |
| 20% | 89.83% | 89.85% | 89.82% | 20% | 74.43% | 74.37% | 74.38% |
| 25% | 89.90% | 89.90% | 89.92% | 25% | 74.48% | 74.43% | 74.44% |
| 30% | 89.98% | 89.96% | 90.05% | 30% | / | 74.46% | 74.48% |
| 35% | 90.04% | 90.02% | 90.09% | 35% | / | 74.48% | 74.50% |
| 40% | 90.10% | 90.06% | 90.15% | 40% | / | 74.51% | 74.55% |
| 45% | 90.17% | 90.10% | 90.24% | 45% | / | 74.53% | 74.60% |
| 50% | 90.21% | 90.14% | 90.35% | 50% | / | 74.55% | 74.62% |

obtain another gain of 0.27%, 0.42%, and 0.52% at maximum percent. Learning from Table 4, it is rational to summarize that HDA can produce massive and high-quality data as HDA-generated data can help to improve the performance of models consistently.

What's more, an interesting phenomenon can be observed that when at smaller percent ($< 25\%$), HybridSet usually performs the worst among three HDA models, whereas at larger percent ($\geq 25\%$), HybridSet consistently performs the best among the three datasets. Therefore, we can conclude that firstly, despite its simplicity, HDA can generate massive and high-quality documents to improve the performance of models; Secondly, HAN with HybridSet shows superiority when a large number of samples added to training set, which indicates that hierarchical information may need more data for models to capture.

## VI. DISCUSSION

Previous work is creative but often complex. And as far as we are concerned, EDA may be the most similar method to our work for its simplicity both on theory and implementation. Based on the results of section IV and V, there are three major differences between HDA and EDA, which can be concluded as follows:

- HDA is a highly attention-based technique, whereas EDA is a highly randomization-based technique.
- The augmented data are treated as independent samples for HDA, whereas treated as attachments to the original data for EDA. As a result, HDA extends the size of datasets, and EDA extends the size of samples.
- HDA augments texts at word-level and sentence level respectively, whereas EDA merely augments texts on word-level. Hence HDA can make full use of the hierarchical structure of texts and guarantee the diversity of new data.

The flaw of randomization-based mechanism is that it is prone to destroy the information of original data. Specifically, it strictly treats each word in sentence as equally important, regardless of semantic information and differentiations on each word's contribution to classification. What's more, due to the randomization-based mechanism, parameter $\alpha$ cannot be set to a large value since it may dramatically destroy the information of original data, whereas small value of $\alpha$ means high similarity between original data and EDA-generated data. To sum up, EDA is an unsatisfying technique since it is hard to produce high-quality data.

Compared with EDA, the proposed HDA has several advantages as follows:

- Avoid information redundancy by distilling the important information from texts
  HDA distills the essential information by extracting the most important words/sentences from texts with attention mechanism, rather than EDA repeatedly produces similar data by change few words in a sentence. Hence, HDA can be very suitable for long-text datasets, and be friendly to model training and data storage.
- Replace randomization-based mechanism with attention mechanism
  Based on attention mechanism, HDA can effectively detect the importance of words/sentences with regard to corresponding documents and selectively distilling important words/sentences to generate new data without changing the topic of documents. Moreover, HDA can also produce massive new data as it is very generic to documents (HDA can augment documents only with the request of having multiple sentences for SentAtt or long sentences for WordAtt). As a comparison, EDA is a highly randomization-based mechanism, and augments texts completely in a random way, which leads to unreliable and uncontrolled data.
- Simplicity
  On the basis of HAN models, HDA only adds a simple operation to extract sentences/words from documents to generate augmented documents using attention values. Thus it does not need to write extra complicated codes. What's more, HDA is also free of requiring any knowledge of a language such as thesaurus and reduces computation costs by a large margin. Hence, HDA can augment texts much faster than EDA.

Additionally, in order to have a more comprehensive understanding of HDA, we also make a broad comparison among HDA and several existing methods, which listed in Table 5.

**TABLE 5. A brief comparison among HDA and several existing methods.**

| Method | Augmenting Level | Implementation Complexity | Extra Support |
|---|---|---|---|
| HDA | **Word and Sentence** | **Very Simple** | **No** |
| Vae+Dis | Sentence | Complex | **No** |
| CA | Word | Complex | Yes, need extra corpus |
| EDA | Word | Simple | Yes, need thesaurus |

Table 5 denotes that there are three advantages of HDA compared to the existing methods as follows:1) it can augment texts on two different levels; 2) Due to its simplicity on theory and implementation, HDA is easy for practice; 3) HDA does not need extra support.

Despite its advantages compared to the existing methods, the proposed HDA still has several flows as follows: 1) the SentAtt and WordAtt work independently with each other, thus HDA does not augment texts at both levels simultaneously; 2) the core of HDA is to summarize the information of texts at both levels via attention mechanism, thus it does not learn to create new data by LMs (Vae+Dis), instead it just condenses the information; 3) SentAtt is not applicable for short-text datasets since their samples mainly have only one sentence.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we firstly propose a novel data augmentation named HDA for text classification, which can augment texts hierarchically and generate new samples by distilling the most important contents from texts based on attention mechanism. Two different augmented datasets, WordSet and SentSet can be created by HDA. The results reveal that HDA can produce massive and high-quality data for training. What's more, HybridSet, the combination of WordSet and SentSet, can further improve the models' performance when a large number of new samples used for training.

For future work, although we utilize WordAtt and SentAtt to augment texts hierarchically, these two methods work independently so far. Thus we plan to jointly utilize SentAtt and WordAtt to augment texts at both levels simultaneously. For instance, we can first use SentAtt to extract the most important sentences, and then use WordAtt to further augment these sentences. Moreover, considering that HDA is evaluated only on text classification task in this paper, we also plan to apply HDA to other NLP tasks like machine translation and question answering.

## REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Mar. 2014.

[2] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[3] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 2096–2104.

[4] Q. Xie, Z. Dai, E. Hovy, M. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," 2019, *arXiv:1904.12848*. [Online]. Available: https://arxiv.org/abs/1907.04658

[5] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," 2018, *arXiv:1805.06201*. [Online]. Available: https://arxiv.org/abs/1805.06201

[6] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. NIPS*, Montreal, QC, Canada, 2015, pp. 649–657.

[7] P. Lewis, L. Denoyer, and S. Riedel, "Unsupervised question answering by cloze translation," 2019, *arXiv:1906.04980*. [Online]. Available: https://arxiv.org/abs/1906.04980

[8] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. NIPS*, Montreal, QC, Canada, 2015, pp. 1684–1692.

[9] S. Ruder. (2018). *Research Directions at AYLIEN in NLP and Transfer Learning*. [Online]. Available: http://blog.aylien.com/research-directions-at-aylien-in-nlp-and-transfer-learning/

[10] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition," in *Proc. ICASSP*, Shanghai, China, 2016, pp. 5755–5759.

[11] Z. Yang, Z. Hu, R. Salakhutdinov, T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," 2017, *arXiv:1702.08139*. [Online]. Available: https://arxiv.org/abs/1702.08139

[12] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114

[13] J. Ian Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[14] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin, "Distilling task-specific knowledge from BERT into simple neural networks," 2019, *arXiv:1903.12136*. [Online]. Available: https://arxiv.org/abs/1903.12136

[15] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," 2017, *arXiv:1703.00955*. [Online]. Available: https://arxiv.org/abs/1703.00955

[16] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," 2019, *arXiv:1901.11196*. [Online]. Available: https://arxiv.org/abs/1901.11196

[17] 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 *Proc. EMNLP*, Lisbon, Portugal, 2015, pp. 2557–2563.

[18] R. Sennrich, B. Haadow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. ACL*, Berlin, Germany, 2016, pp. 86–96.

[19] K. Kafle, M. Yousefhussien, and C. Kanan, "Data augmentation for visual question answering," in *Proc. 10th Int. Conf. Natural Lang. Gener.*, 2017, pp. 198–202.

[20] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473

[22] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI*, Austin, TX, USA, 2015, pp. 2267–2273.

[23] J. Abreu, L. Fred, D. Macêdo, and C. Zanchettin, "Hierarchical attentional hybrid neural networks for document classification," 2019, *arXiv:1901.06610*. [Online]. Available: https://arxiv.org/abs/1901.06610

[24] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," 2018, *arXiv:1808.09381*. [Online]. Available: https://arxiv.org/abs/1808.09381

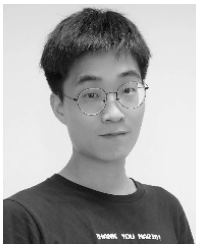
[25] M. Fadaee, A. Bisazza, and C. Monz, "Data Augmentation for Low-Resource Neural Machine Translation," in *Proc. ACL*, Vancouver, BC, Canada, vol. 2, 2017, pp. 567–573.

[26] A. G. Howard, "Some improvements on deep convolutional neural network based image classification," 2013, *arXiv:1312.5402*. [Online]. Available: https://arxiv.xilesou.top/abs/1312.5402

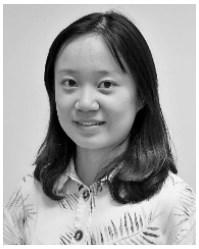
[27] Q. Diao, M. Qiu, C. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation," in *Proc. ACM SIGKDD (KDD)*, New York, NY, USA, 2014, pp. 193–202.

[28] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*. [Online]. Available: https://arxiv.org/abs/1607.04606
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[30] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," 2015, *arXiv:1507.06228*. [Online]. Available: https://arxiv.org/abs/1507.06228

**SHUJUAN YU** received the B.E. degree from the Harbin Institute of Technology, Harbin, China, in 1989, and the M.S. degree from Southeast University, Nanjing, China, in 1995. She has been an Associate Professor and a Master Tutor with the College of Electronic and Optical Engineering, Nanjing University of Posts and Telecommunications, since 2007. And her research interests are in the fields of adaptive signal processing, deep learning, and intelligent big data processing.



**JIE YANG** received the B.E. degree from the Zhejiang University of Media and Communications, in 2017. He is currently pursuing the master's degree with the Nanjing University of Posts and Telecommunications, under the supervision of Prof. Yu. His research interests are deep learning, natural language processing, and data analytics.



**DANLEI LIU** received the B.E. degree from the New York Institute of Technology and the Nanjing University of Posts and Telecommunications, in 2019, where she is currently pursuing the master's degree under the supervision of Prof. Yu. Her main research directions are deep learning, knowledge graph, and natural language processing.

**RUNQI LI** received the B.E. degree from the Nanjing University of Posts and Telecommunications, in 2018. She is currently pursuing the dual master's degrees with the Queen Mary University of London and the Nanjing University of Posts and Telecommunications, under the supervision of Prof. Yu. Her main research directions are machine learning, data mining, and natural language processing.

**YUN ZHANG** received the M.S. and Ph.D. degrees from the Nanjing University of Posts and Telecommunication, Nanjing, China, in 2005 and 2011, respectively. She had been a Visiting Scholar with the State University of New York at Binghamton in 2017. She is currently a Lectorate with the College of Electronic and Optical Engineering, Nanjing University of Posts and Telecommunications. Her research interests are in the fields of blind channel equalization, machine learning, and wireless communications.

**SHENGMEI ZHAO** was born in 1968. She has been a Professor and a Ph.D. Tutor with the School of Communication and Information Engineering, Nanjing University of Posts and Telecommunication. At present, she mainly studies intelligent signal processing technologies, data mining, and machine learning.