

# A Text Data Augmentation Approach for Improving the Performance of CNN

Muhammad Abulaish, *SMIEEE*  
Department of Computer Science  
South Asian University, New Delhi, India  
Email: abulaish@ieee.org

Amit Kumar Sah  
Department of Computer Science  
South Asian University, New Delhi, India  
Email: amitjanakpuri@hotmail.com

**Abstract**—Deep learning is an emerging research area in the field of machine learning and its fascinating accuracy has attracted many researchers to apply it in various domains, including computer vision and natural language processing. Traditional machine learning approaches require to invest a good amount of time in feature engineering and related tasks. Deep learning, on the other hand, does not require to define features explicitly; instead, it aims to learn different representations from data automatically. However, it needs large enriched corpus to train deep classification models properly. Overfitting is another challenging issue, which needs enriched corpus. In this paper, we propose a data augmentation approach, which combines  $n$ -grams and LDA techniques to identify class-specific phrases to enrich the underlying corpus. We have evaluated the performance of the convolutional neural network on both original and augmented corpus, and it is found that the augmented corpus has lower variance and better validation accuracy in comparison to the original corpus. The proposed data augmentation approach seems very useful for the domains with small data corpus to train deep learning models.

**Index Terms**—Machine Learning, Deep Learning, Convolutional Neural Network, LDA, Data Augmentation.

## I. INTRODUCTION

The invention of the Web 2.0 has given birth to many online social networking sites and people from different age groups are active over these platforms for their day-to-day activities, including entertainment, news sharing, event updates, etc. As a result, huge amount of electronic data, mainly unstructured or semi-structured, is being generated and resulted in *information overload* problem. Besides natural language processing and information retrieval and extraction techniques, many researchers have used machine learning techniques to analyze unstructured textual data automatically for varied purposes, including document classification, document summarization, named-entity extraction, relation mining, data curation, etc. Machine learning algorithms aim to learn patterns from data, rather than programming everything explicitly, so that every new entity instance could be assigned a class automatically. However, traditional machine learning techniques involve lots of feature extraction and engineering tasks.

Deep learning, on the other hand, does not require features to be explicitly defined; rather, it aims to learn different representations from data automatically. Recently, deep

learning techniques have emerged as effective classification models and they are very successful in many domains. Deep learning models consists of multiple processing layers to learn representation of data at different levels of granularity to discover intricate structures in large datasets by using the backpropagation algorithms [1]. Though the first big deep learning breakthrough was in the field of NLP in 2011 [2], it became more popular in the year 2012, when deep learning based convolutional neural network (CNN), AlexNet, won the prestigious ImageNet competition reducing the error rate remarkably from 25.8% to 16.4%. Since then, researchers have been working continuously to make improvements in this field and exploration of its application areas.

Although deep learning techniques are widely lauded due to their significant classification performance, some of the main challenges faced by the deep learning models are the requirement of large corpora, overfitting, and hyper-parameters selection. In many domains, very few data are generated, and small corpus is one of the major concerns for deep learning models because they need quite a huge corpus to learn representations of data. In addition, if a corpus is not enriched, then the deep learning models are susceptible to overfitting.

In this paper, we propose a data augmentation approach which combines  $n$ -grams and Latent Dirichlet Allocation (LDA) techniques to identify class-specific phrases for enrichment of the underlying corpus. Data augmentation is one of the key approach to combat overfitting problem in deep learning models. The performance of a classifier largely depends on the training datasets, and generally real-world datasets are not linearly separable. For example, in a reviews corpus, same words may be present in both positive and negative review documents. Therefore, in line with the stratified cross-validation approach, we propose to enrich corpus based on the class labels of the individual instances. For example, in a review documents corpus, positive review documents can be augmented with positive phrases and negative review documents with negative phrases, resulting in increased intra-class similarity and thereby facilitating classifiers to learn discriminating patterns. The proposed data augmentation approach augments individual instances with an aim to enrich the dataset with more contextual

phrases containing the keywords related to the respective class. The proposed approach is evaluated over four Amazon’s review datasets using CNN, considering the sentiment classification as a logistic regression task, which is similar to the binary classification problem. The performance of CNN over normal (non-augmented) and augmented datasets is compared and it is found that CNN performs significantly better on augmented datasets.

## II. RELATED WORKS

A number of approaches for data augmentation exist in literature. In image processing domain, addition of flipped, enlarged, rotated, and transformed images in the corpus has resulted in improved classifiers’ accuracy [3], [4]. Similarly, in text information processing domain, addition of synonyms in the corpus has been done by some of the researchers [5]. In short texts, such as reviews and tweets where number of words are very few, it is difficult for classifiers to learn meaningful patterns. In such cases, data augmentation seems essential to improve generalization ability of the deep learning models.

The researches presented in [3], [6] are some of the remarkable works in pattern recognition using neural networks that have used data augmentation. In [5], the authors performed data augmentation using English thesaurus and evaluated using deep learning models. In [4], the authors discussed the effectiveness of data augmentation mechanism for images in deep learning models. In [7], the authors used data augmentation for handwritten words and line recognition using a CNN-LSTM network.

Apart from classification, several other works have been done in the field of text data augmentation. In [8], the authors explored several sense-based strategies for web search query expansion, where a user query is reformulated by adding new meaningful terms with similar significance [9]. They augmented words with the same sense using WordNet semantic network, instead of using just synonyms. In [10], the authors discussed a method for query expansion using query logs. In contrast to the existing works on textual data augmentation, our proposed approach augments textual data using the context, class labels, and importance of the words identified through LDA.

## III. PROPOSED TEXT DATA AUGMENTATION APPROACH

In this section, we present the work-flow of the proposed text data augmentation approach, which is shown in figure 1. A detailed description of the individual processes are presented in the following sub-sections.

### A. Data Pre-processing

Text documents generally possess a high degree of variation from usual grammatical structures. They contain mostly innovated spellings developed by the users due to character limitation, informal writing, etc. Simple pre-processing is not enough for such data, as it may lead to semantic loss. Different pre-processing tasks applied to the

text corpus considered in this study are mainly stop-words, URLs and hashtag symbols removal, resolving elongated words, emoticons handling, resolving contractions, stemming, and lemmatization.

### B. Corpus Splitting

In this section, we discuss the process to split a corpus into class-specific sub-corpus. In this study, we splitted the reviews corpus ( $X$ ) into two sub-corpus – *positive reviews corpus* ( $X_P$ ) and *negative reviews corpus* ( $X_N$ ) based on the star ratings of the individual reviews. The positive reviews corpus contains the reviews having star rating values greater than or equal to 3, whereas negative reviews corpus contains the reviews having star rating values less than 3. Algorithm 1 (CorpusSplitting) describes the corpus splitting process formally.

---

#### Algorithm 1: CorpusSplitting( $X$ )

---

**Input** : Pre-processed reviews corpus  $X$   
**Output**: Positive reviews corpus  $X_P$  and negative reviews corpus  $X_N$

```

1  $X_P \leftarrow \phi, X_N \leftarrow \phi;$ 
2 foreach  $review$  in  $X$  do
3   if  $rating(review) \geq 3$  then
4      $X_P \leftarrow X_P \cup \{review\};$ 
5   end
6   else
7      $X_N \leftarrow X_N \cup \{review\};$ 
8   end
9 end
10 return  $X_P, X_N$ 

```

---

### C. Keywords Extraction

In this section, we discuss the keywords extraction process using LDA, which is a generative probabilistic model and it is generally used to detect underlying topics in text documents. In this model, documents are represented as random mixtures over latent topics, where each topic is characterized by the distribution of the relevant words [11]. We applied LDA on both positive and negative reviews corpuses separately, keeping the number of topics as 1. We sorted the LDA results in descending order based on the relevance scores of the topic words and considered top- $k$  words from each positive and negative reviews sub-corpus to generate the list of *positive reviews keywords* ( $K_P$ ) and *negative reviews keywords* ( $K_N$ ), respectively. In this study, we have considered the value of  $k$  as 500. Algorithm 2 (KeywordsExtraction) describes the keywords extraction process formally.

### D. Data Augmentation

This section presents a brief detail about the data augmentation process. To this end, we have used the concept of  $n$ -gram, which gives a good probabilistic estimate of the contextual words for any given word of a document. An  $n$ -gram is a contiguous sequence of  $n$  elements (characters, words, etc.) from a given sequence of texts, and based on the values of  $n$  as 1, 2, or 3, it is termed as *unigram*, *bigram*, or *trigram*, respectively. In this study, we have

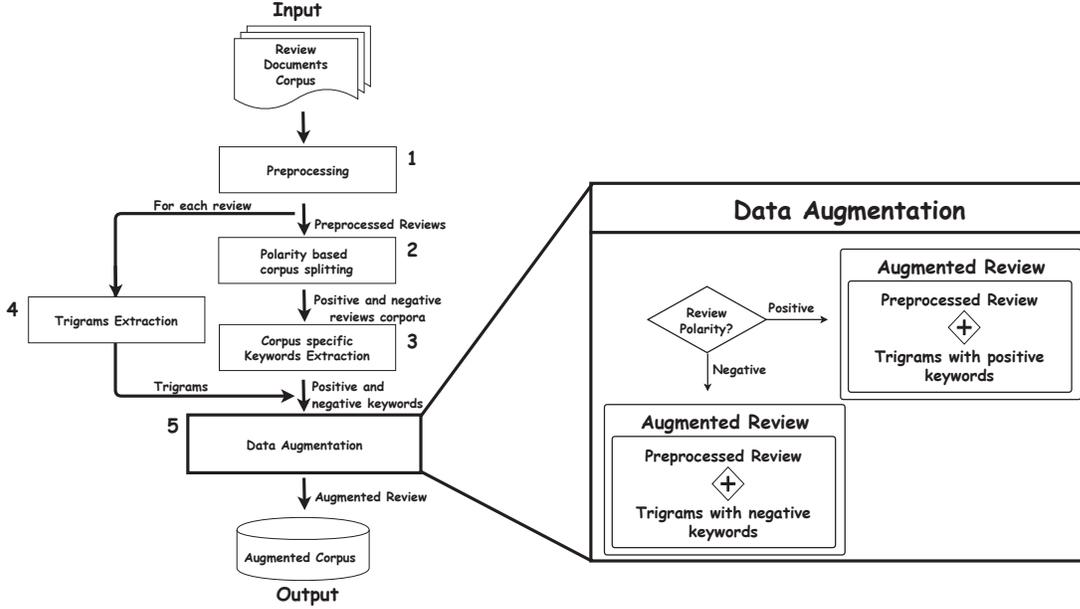


Fig. 1: Work-flow of the proposed text data augmentation approach

---

### Algorithm 2: KeywordsExtraction( $X_P, X_N, k$ )

---

**Input** : Positive reviews corpus  $X_P$ , negative reviews corpus  $X_N$ , and an integer value  $k$   
**Output**:  $K_P$ : top- $k$  keywords from  $X_P$ ,  $K_N$ : top- $k$  keywords from  $X_N$

```

1  $K_P \leftarrow \phi, K_N \leftarrow \phi;$ 
2  $KW \leftarrow \text{LDA}(X_P, \text{num\_of\_topics}=1);$ 
3 foreach  $keyword$  in  $KW$  do
4   if  $keyword$  is in top- $k$  words of  $KW$  then
5      $K_P \leftarrow K_P \cup \{keyword\};$ 
6   end
7 end
8  $KW \leftarrow \text{LDA}(X_N, \text{num\_of\_topics}=1);$ 
9 foreach  $keyword$  in  $KW$  do
10  if  $keyword$  is in top- $k$  words of  $KW$  then
11     $K_N \leftarrow K_N \cup \{keyword\};$ 
12  end
13 end
14 return  $K_P, K_N$ 

```

---

generated the trigrams of each review document. If the review document belongs to the positive reviews corpus then we augment it with all of its trigrams that contain at least one keyword from the list of *positive reviews keywords*. Similarly, if the review document belongs to the negative reviews corpus then we augment it with all of its trigrams that contain at least one keyword from the list of *negative reviews keywords*. Finally, both augmented positive reviews corpus and augmented negative reviews corpus are combined together to obtain the augmented reviews corpus. Algorithm 3 (CorpusAugmentation) presents this data augmentation process formally.

Mathematically, if  $X$  represents the normal (non-augmented) reviews corpus and  $x_i$  is the  $i^{th}$  review document in  $X$ , then  $X$  can be represented as  $X = \{x_1, x_2, \dots, x_n\}$ , where  $n$  is the total number of reviews in  $X$ . Similarly, each review  $x_i$  can be represented as

---

### Algorithm 3: CorpusAugmentation( $X_P, X_N, K_P, K_N$ )

---

**Input** : Positive reviews corpus  $X_P$ , negative reviews corpus  $X_N$ , list of positive reviews keywords  $K_P$ , and list of negative reviews keywords  $K_N$   
**Output**: Augmented corpus  $AX$

```

1  $AX_P \leftarrow \phi;$ 
2 foreach  $review$  in  $X_P$  do
3    $review\_trigram \leftarrow \text{ngram}(review, n = 3);$ 
4   foreach  $trigram$  in  $review\_trigram$  do
5     if  $trigram$  contains any keyword from  $K_P$  then
6        $review \leftarrow review + trigram;$ 
7     end
8   end
9    $AX_P \leftarrow AX_P \cup \{review\};$ 
10 end
11  $AX_N \leftarrow \phi;$ 
12 foreach  $review$  in  $X_N$  do
13    $review\_trigram \leftarrow \text{ngram}(review, n = 3);$ 
14   foreach  $trigram$  in  $review\_trigram$  do
15     if  $trigram$  contains any keyword from  $K_N$  then
16        $review \leftarrow review + trigram;$ 
17     end
18   end
19    $AX_N \leftarrow AX_N \cup \{review\};$ 
20 end
21  $AX \leftarrow AX_P \cup AX_N;$ 
22 return  $AX$ 

```

---

$x_i = \{w_1, w_2, \dots, w_l\}$ , where  $l$  is the number of words in  $x_i$ .

Algorithm 1 (CorpusSplitting) splits the corpus  $X$  into two sub-corpora – *positive reviews corpus* ( $X_P$ ) and *negative reviews corpus* ( $X_N$ ) such that  $X = X_P \cup X_N$ , where  $X_P$  is the set of all positive reviews and  $X_N$  is the set of all negative reviews. A review document  $x_i \in X_P$  if  $rating(x_i) \geq 3$ , and  $x_i \in X_N$  if  $rating(x_i) < 3$ .

Algorithm 2 (KeywordsExtraction) extracts keywords from each sub-corpus  $X_P$  and  $X_N$ . Let the keywords extracted from  $X_P$  are represented as  $K_P =$

TABLE I: Statistics of the datasets

Corpus	#Total reviews	#Positive reviews	#Negative reviews
Amazon Instant Video	37,126	33,523	3,603
Automotive	20,473	19,325	1,148
Digital Music	64,706	58,905	5,801
Office Products	53,258	50,402	2,856

$\{pk_1, pk_2, \dots, pk_m\}$  and that from  $X_N$  are represented as  $K_N = \{nk_1, nk_2, \dots, nk_m\}$ , where  $m$  is the maximum number of keywords.

Algorithm 3 (CorpusAugmentation) returns augmented corpus  $AX$  such that  $AX = AX_P \cup AX_N$ , where  $AX_P$  is the augmented positive reviews corpus and  $AX_N$  is augmented negative reviews corpus. Let each review document  $x_i$  is represented as a set of *trigrams*  $x_i = \{tr_1, tr_2, \dots, tr_q\}$ , where  $q$  is the total number of trigrams in  $x_i$ . Each augmented document  $Ax_i \in AX_P$  is represented as  $Ax_i = \{w_1, w_2, \dots, w_l, tr_1, tr_2, \dots, tr_j\}$ , where  $w_1, w_2, \dots, w_l$  are the original words of  $x_i$ , and  $tr_1, tr_2, \dots, tr_j$  are the *trigrams* of  $x_i$  that contain at least one keyword from  $K_P$ . Similarly, each augmented document  $Ax_i \in AX_N$  is represented as  $Ax_i = \{w_1, w_2, \dots, w_l, tr_1, tr_2, \dots, tr_k\}$ , where  $w_1, w_2, \dots, w_l$  are the original words of  $x_i$ , and  $tr_1, tr_2, \dots, tr_k$  are the *trigrams* of  $x_i$  that contain at least one keyword from  $K_N$ .

#### IV. EXPERIMENTAL SETUP AND RESULTS

In this section, we present our experimental setup and evaluation of the effectiveness of the proposed data augmentation approach. All experiments were performed on a machine with 2.0 GHz Intel Core i3 processor and 16G RAM. Our data augmentation model was implemented in Keras, which is a high-level neural networks API written in Python and capable of running on top of TensorFlow, CNTK, or Theano.

##### A. Datasets

We have used publicly available Amazon’s datasets [12], which contain user reviews on different Amazon’s products and services. All datasets comprise of reviews as well as related metadata, such as user name, overall rating, time, date, etc. Although instances in these datasets are labeled with five different classes, we modified them to binary class based on the star ratings of the individual reviews. All reviews with star rating 1 or 2 were labeled as negative reviews, whereas reviews with star ratings 3, 4 or 5 were labeled as positive reviews. Table I presents the statistics of the modified datasets.

##### B. Architecture of the CNN Model

In this section, we discuss the classification mechanism used to verify the effectiveness of the augmented corpus. Since, the reviews corpus is divided into two sub-corpus namely, *positive reviews corpus* and *negative reviews corpus*, we have considered the problem as a logistic regression task, which is similar to the binary classification problem.

To this end, we have used a 9-layer Convolutional Neural Network (CNN) architecture, having 6 convolutional layers and 3 fully-connected layers. At every convolutional layer, we have used 256 filters. There are 1024 neurons in the immediate two dense layers after the convolutional layers and single neuron in the 9<sup>th</sup> layer, because we have formulated the classification problem as a logistic regression task.

Tables II and III present the details of the convolutional layers and fully-connected layers, respectively used in our experiment.

TABLE II: Description of the convolutional layers

Layers	#Kernels	Kernel size	Pool size
1	256	5	2
2	256	5	2
3	256	3	N/A
4	256	3	N/A
5	256	3	N/A
6	256	3	3

TABLE III: Description of the fully-connected layers

Layers	#Output units
7	1024
8	1024
9	1

*Xavier Glorot* initialization is used to assign initial weights and *adam* is used as an optimizer in our model. To reduce overfitting effect, our model has *dropout* in between fully-connected layers with probability 0.5. We have also used *l2* regularizer to reduce the overfitting effect with value of  $\lambda$  as 0.01. *L2* regularization is given by equation (1), where  $m$  is the number of training examples and the value of  $\lambda$  is determined empirically, keeping in mind that increasing it reduces overfitting but increases the bias.

Rectified Linear Unit (ReLU) is the activation function, which is used throughout the model, except in the output layer where we have used sigmoid activation function. ReLU is given by equation (2), where  $x$  is the input and  $max$  refers to maximum. Since we are using the proposed CNN model for logistic regression task, we have considered sigmoid activation function in the last layer. The Sigmoid activation function is defined in equation (3), where  $x$  is the input value.

$$\frac{\lambda}{2m} \times \sum ||w||^2 \quad (1)$$

$$ReLU(x) = max(0, x) \quad (2)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

We have represented each review document as an  $n_w \times d_w$  matrix, where  $n_w$  is the maximum number of words present in a review document and  $d_w$  is the embedding size

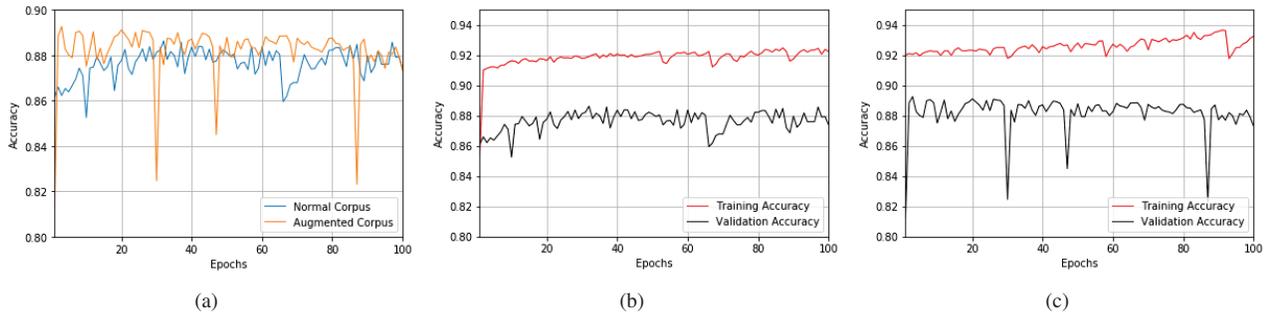


Fig. 2: Evaluation results on *amazon instant video* dataset: (a) validation accuracy (normal vs augmented corpus), (b) training vs validation accuracy (normal corpus), (c) training vs validation accuracy (augmented corpus)

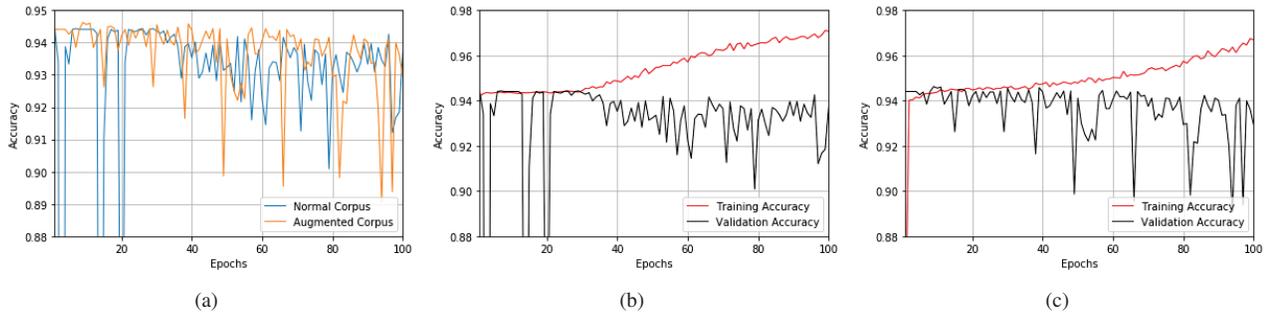


Fig. 3: Evaluation results on *automotive* dataset: (a) validation accuracy (normal vs augmented corpus), (b) training vs validation accuracy (normal corpus), (c) training vs validation accuracy (augmented corpus)

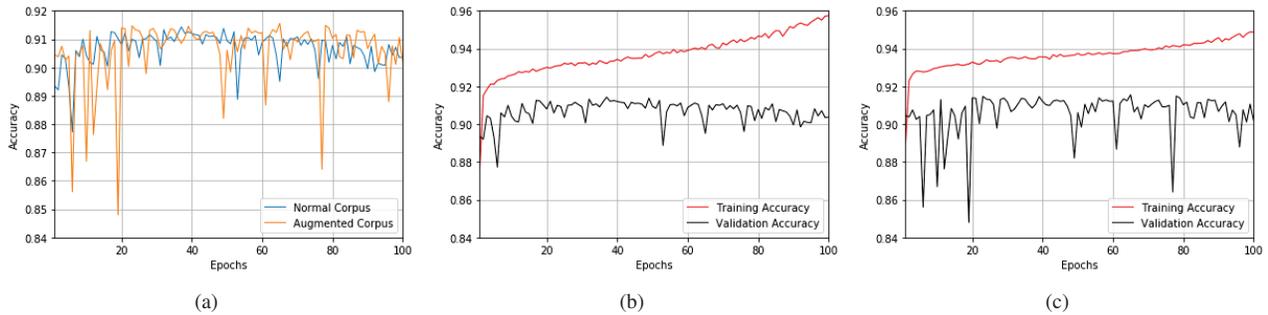


Fig. 4: Evaluation results on *digital music* dataset: (a) validation accuracy (normal vs augmented corpus), (b) training vs validation accuracy (normal corpus), (c) training vs validation accuracy (augmented corpus)

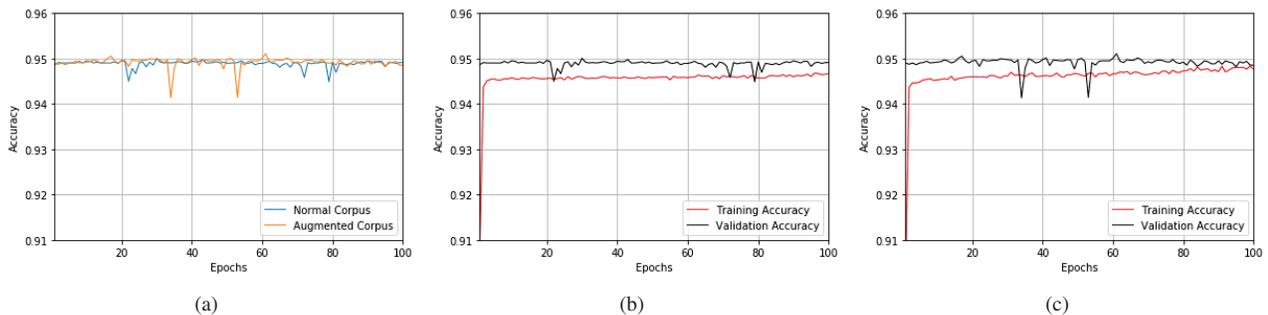


Fig. 5: Evaluation results on *office products* dataset: (a) validation accuracy (normal vs augmented corpus), (b) training vs validation accuracy (normal corpus), (c) training vs validation accuracy (augmented corpus)

TABLE IV: Corpus-wise maximum number of words chosen for the classification tasks

Corpus	Maximum no. of words	
	Normal	Augmented
Amazon Instant Video	60	80
Automotive	60	80
Digital Music	100	150
Office Products	60	80

of each word. We have used pre-trained GloVe 300 dimensional embeddings with 840 billion tokens. We have chosen the maximum number of words depending on the corpus and considering whether we are performing classification on normal corpus or augmented corpus. Details about the maximum number of words chosen for the classification tasks are presented in table IV. Input size or maximum number of words for different datasets (normal as well as augmented corpuses) is different during the classification process because of the fact that different datasets have different average review length. Choosing maximum words is totally empirical and it can be seen as a hyperparameter of the CNN. To make our algorithm working properly, for all those words of a corpus having no embeddings in *GloVe Embeddings*, we have generated embeddings using the *Word2Vec skip-gram model* on the corpus locally.

### C. Results

In this section, we discuss the results obtained using CNN model over both normal and augmented datasets. We deployed our CNN model on both normal and augmented versions of the datasets mentioned in table I for 100 epochs. For each epoch, we used 80% data for training and remaining 20% data for validation. We recorded the results obtained after 100 epochs and visualized them using the plots shown in figures 2–5. The following paragraphs summarize some of the key observations from these figures:

- Figures 2(a), 3(a), 4(a) and 5(a) show comparatively higher validation accuracy over the augmented corpuses in comparison to the normal corpuses. This suggests that an augmented corpus is more effective than the normal corpus, with respect to the proposed CNN model.
- It can be observed from the figures 2(b), 3(b), 4(b) and 5(b) that the classifier shows high variance over the normal corpuses, i.e., there is significant variation between the training and validation accuracies. On the other hand, it can be observed from the figures 2(c), 3(c), 4(c) and 5(c) that the variation between the training and validation accuracies is low over the augmented corpuses. It can also be observed that the

### V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a text data augmentation approach using LDA and  $n$ -grams techniques. The proposed

rate of overfitting is reduced when the experiment is performed over the augmented corpuses.

approach seems interpretable and it boosts the performance of the CNN. Instead of applying LDA on the whole corpus, its application on the class-specific instances of the text corpus is proved to make the corpus more discriminative. High variance was observed in normal corpus in comparison to the augmented corpus, which reflects the significance of the proposed data augmentation approach. The augmented corpus could be very useful for deep learning models that learn patterns from the data itself. The proposed text data augmentation approach also seems very useful for the domains having small-sized text documents (e.g., tweets) because it increases the document size to avoid the problem of information scarce. The proposed approach is also generic in nature because it can be used to augment any labelled text corpus. Application of the proposed approach to handle class imbalance problem seems one of the promising research problems in the field of text information processing.

### REFERENCES

- [1] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 5 2015.
- [2] A. r. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, “Deep belief networks using discriminative features for phone recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5060–5063, May 2011.
- [3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, pp. 2278–2324, 1998.
- [4] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017.
- [5] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *NIPS*, pp. 649–657, 2015.
- [6] N. Jaitly and G. E. Hinton, “Vocal tract length perturbation (vtl) improves speech recognition.”
- [7] C. Wigginton, S. Stewart, B. Davis, B. Barrett, B. Price, and S. Cohen, “Data augmentation for recognition of handwritten words and lines using a cnn-lstm network,” in *2017 14th IAPR International Conference on Document Analysis and Recognition*, vol. 01, pp. 639–645, Nov 2017.
- [8] R. Navigli and P. Velardi, “An analysis of ontology-based query expansion strategies,” 01 2003.
- [9] H. K. Azad and A. Deepak, “Query expansion techniques for information retrieval: a survey,” *CoRR*, vol. abs/1708.00247, 2017.
- [10] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, “Probabilistic query expansion using query logs,” in *Proceedings of the 11th International Conference on World Wide Web, WWW’02*, pp. 325–332, 2002.
- [11] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
- [12] R. He and J. McAuley, “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering,” in *Proceedings of the 25th International Conference on World Wide Web, WWW’16*, pp. 507–517, 2016.
- [13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Nevada, United States* [13], pp. 3111–3119.