Abstract

Text Classification is a central task of Natural Language Processing. A good text classification model comes with the prerequisite of good text classification data. Taking into account the augmentation techniques used for creating new data in images, a novel approach to data augmentation was found in EDA augmentation. The aim of this work is to implement the AEDA augmentation and compare its performance with the original and the EDA augmented datasets. The big promise of AEDA is that with relatively simpler augmentation methods it can achieve nearly the same amount of accuracy as the EDA augmented dataset on the same base model. We have implemented both the EDA and the AEDA augmentations in our work to find out whether AEDA can live up to the hype or not. Surprisingly, our observations have been along the positive lines and we have observed that the AEDA augmentation delivers on its promise.

1. Introduction

In natural language processing, text classification is a crucial task. Machine learning and deep learning have achieved high accuracy on tasks ranging from sentiment analysis to topic classification, but excellent performance is frequently dependent on the number and quality of training data, which may be time-consuming to gather.

If there is insufficient labelled data for training in many machine learning (ML) applications and domains, data augmentation (DA) can be used to increase the performance of machine learning systems.

For the data augmentation in the image classification task, methods like rotating the image, zooming in the image, changing the angles, smearing the image etc. are used to improve the performance of the model and reduce overfitting. However, in the case of text data, the augmentation is less intuitive as compared to image data.

Various methods that can be used to carry out data augmentation include altering elements of the input sequence like word substitution, deletion, insertion and back translation or injecting noise into the input sequence.

One such method is EDA, where they have used Random Synonym replacement, Random Insertion, Random Deletion, and Random Swap. Theoretically, this method suffers from the problem of information loss due to random deletion and substitution.

To address this problem an extremely simple yet effective approach called AEDA (An Easier Data Augmentation) has been proposed. It includes the insertion of various punctuation marks into the input sequence. Because it leaves the word order intact while shifting the words to the right, AEDA retains all of the input information and does not lead the network wrong. In theory, AEDA can be seen as a measure to avoid overfitting in the model.

2. Literature Survey

Wang and Yang (2015) substitute words with their synonyms for classifying tweets

Sennrich et al. (2016) utilized back-translation to train a neural machine translation system using automatically translated data as well as the original human-translated data.

Fadaee et al. (2017) utilized substitution of common words with rare ones, thus providing more context for the rare words,

When training, Hu et al. (2019) and Liu et al. (2020) use reinforcement learning with a conditional language model, which involves attaching the proper label to the input sequence.

Xie et al. (2019) employ data noising, which is similar to our approach with the exception that they use the underscore character as a placeholder or replace terms from the unigram frequency distribution.

Andreas (2020) creates new sentences by replacing sentence fragments from similar categories with each other.

Some researchers have opted for using pre-trained language models such as BERT.

Our approach is really simple to use and does not require any additional information.

3. Methodology

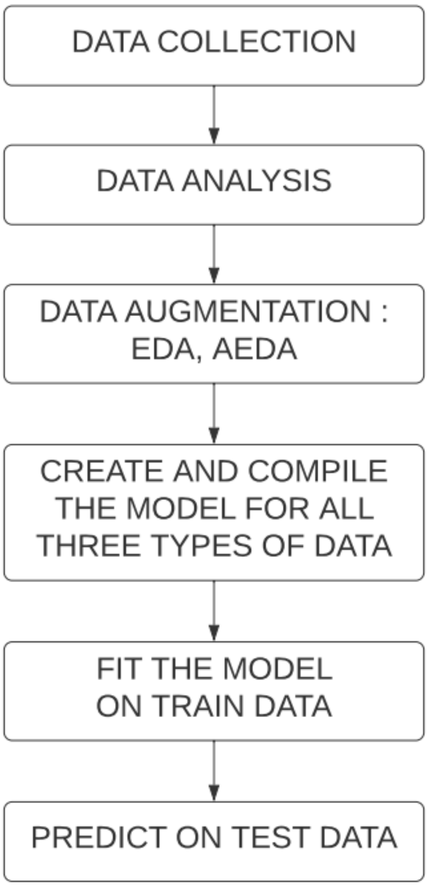

DATA COLLECTION
DATA ANALYSIS
DATA AUGMENTATION : EDA, AEDA
CREATE AND COMPILE THE MODEL FOR ALL THREE TYPES OF DATA
FIT THE MODEL ON TRAIN DATA
PREDICT ON TEST DATA

Figure1. Workflow

3.1. Dataset

Experimentation is performed on the same five datasets as the baseline. The datasets used are CR(Customer Reviews Dataset), PC(Pros and Cons Dataset), SST-2(Standford Sentiment Treebank), and SUBJ(Subjectivity/Objectivity Dataset) and TREC (Question Classification Dataset).

Train and Test sets were not made available by the baseline. So after collecting them we shuffled and divided them into train and test sets with almost the same sizes as mentioned in the baseline. In the CR dataset, we combined all the reviews from the three cited sources. The annotations included multiple target sentiments for each sentence.

Therefore, to convert them into binary classes, we considered a sentence positive if there was no negative sentiment and negative if there was no positive sentiment.

Dataset	No. of Classes	Average sentence length	No. of training samples	No. of test samples	No. of unique words
SST-2	2	19	7791	1821	15771
CR	2	19	4067	451	9048
SUBJ	2	25	9000	1000	22715
TREC	6	10	5452	500	9448
PC	2	7	40000	5806	26090

Table 1. Statistics of the utilized datasets

3.2. EDA

First, we extracted all the words from the sentences and removed the stopwords from them. We have used the in-built nltk stopwords set for this task.

1. Synonym Replacement (SR): We have chosen 10 percent words out of the total number of words in a sentence for synonym replacement randomly.

2. Random Insertion (RI): We have randomly inserted 10 percent words out of the total number of words in a sentence.

3. Random Swap (RS): For 10 percent of the words out of the total number of words in a sentence, we have randomly chosen another word and replaced the two.

4. Random Deletion (RD): We have randomly deleted 10 percent of the words out of the total number of words with a probability of 90 percent in a sentence.

Original Sentence: “ `The Hitachi is made in Malaysia, and looked cheap compared with the Makita, which is made in the USA.″`

After Cleaning: ‟ `the hitachi is made in malaysia and looked cheap compared with the makita which is made in the usa ‟`

EDA Augmentation:

DELETION: “ the made in malasyia and looked compared with the makita which is made in the usa”

SWAP: “the hitachi is made makita malasyia and looked cheap compared with the in which is made in the usa”

SYNONYM REPLACEMENT: “ the hitachi is made in malasyia and seem cheap compared with the makita which is made in the usa ”

INSERTION: “ the hitachi is made in malasyia and looked cheap compared with the follow makita which is made in the usa “

3.3. AEDA

We randomly insert punctuations in each sentence, at the rate of 30% out of the total number of words in each sentence. We perform 4 augmentations for each sentence.

1. The Hitachi is ; made in : Malasyia, , and looked cheap compared ; with ? the Makita, which is ? made in the USA.

2. ? The : Hitachi : is made in Malasyia, and looked : cheap compared with . the Makita, which is made in the USA.

3. The Hitachi is made in Malasyia, and . looked cheap compared , with ; the Makita, which is , made in , the USA.

4. The Hitachi is made in Malasyia, and looked cheap compared with the Makita, which . is made in . the USA.

3.4. Model

We have used a CNN Model, the architecture of which can be seen in the following diagram. We have compiled the model with the adam optimizer, using the categorical cross-entropy as the loss function and categorical accuracy as the metric.

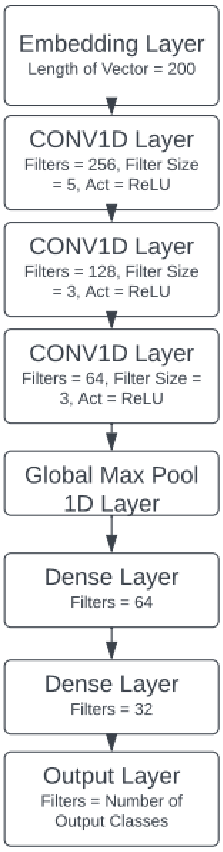
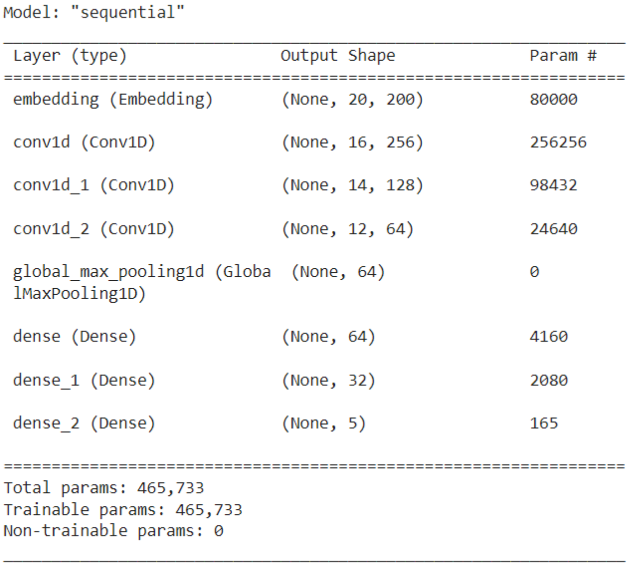

Embedding Layer
Length of Vector = 200
CONV1D Layer
Filters = 256, Filter Size = 5, Act = ReLU
CONV1D Layer
Filters = 128, Filter Size = 3, Act = ReLU
CONV1D Layer
Filters = 64, Filter Size = 3, Act = ReLU
Global Max Pool 1D Layer
Dense Layer
Filters = 64
Dense Layer
Filters = 32
Output Layer
Filters = Number of Output Classes

Figure 2. Model



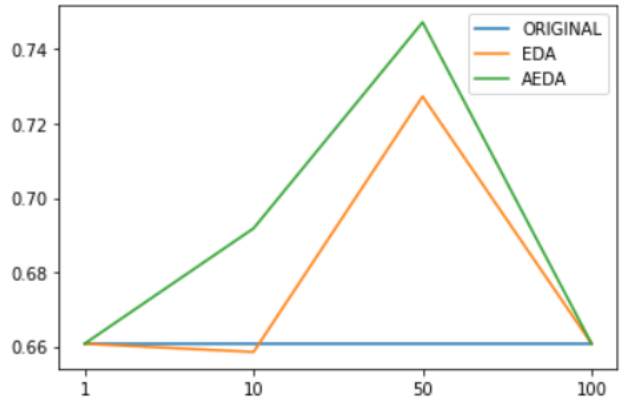
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 20, 200)           80000

 conv1d (Conv1D)             (None, 16, 256)           256256

 conv1d_1 (Conv1D)           (None, 14, 128)           98432

 conv1d_2 (Conv1D)           (None, 12, 64)            24640

 global_max_pooling1d (Globa  (None, 64)               0
 lMaxPooling1D)

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 5)                 165

=================================================================
Total params: 465,733
Trainable params: 465,733
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. Model Summary

4. Results and Discussion

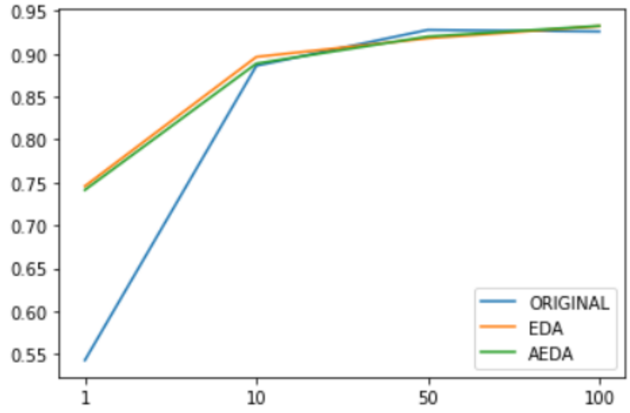
We have trained the model on different percentages of data [1,10,50,100] on the same model.

1. CR



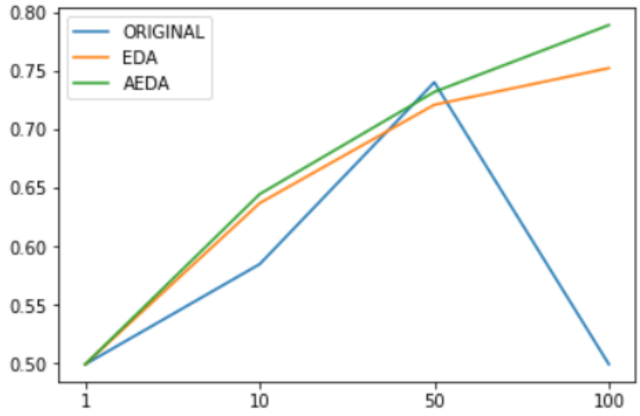
(a) CR dataset performance

2. PC



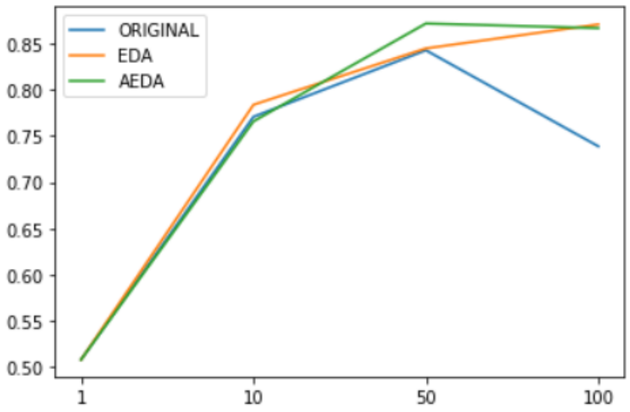
(b) PC Dataset Performance

3. SST-2



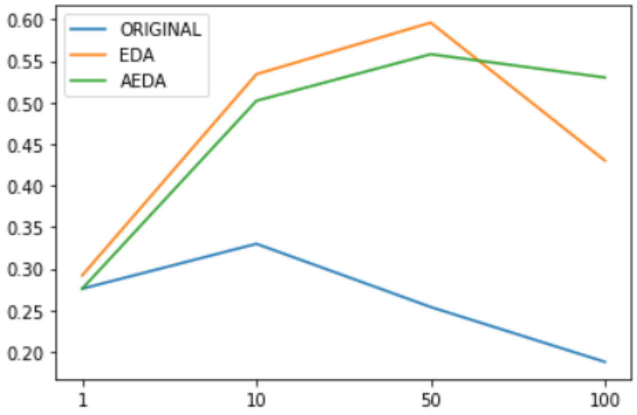
(c) SST-2 Dataset Performance

4. SUBJ



(d) SUBJ Dataset Performance

5. TREC



(e) TREC Dataset Performance

Figure 4. Performance of the model trained on various proportions of the original, EDA-generated, and AEDA-generated training data for five text classification tasks.

5. Conclusion

We can make the following conclusions from the graphs above, firstly that there is not much of a difference between the performance of the CNN model on the augmented dataset as compared to the original dataset when the size of the dataset is large, the reason for this fact can be that for larger datasets there is not a need for augmentation since enough data is available to extract the important features. Also, it can be seen that both EDA and AEDA datasets were able to delay and reduce overfitting as the size of the datasets increased. Lastly, the aim and the big promise of AEDA augmentation was that it can be used to achieve the same level of performance for the model as the

EDA dataset with less and simpler work as compared to EDA. As can be seen, by the graphs, the accuracy for both the augmentations is nearly the same, and thus we can say that AEDA augmentation has proven to be just as effective, under the conditions used by us, as the EDA augmentation with comparatively simpler and less work.

6. References :

[1]. Akbar Karimi Leonardo Rossi Andrea Prati, 2021, AEDA: An Easier Data Augmentation Technique for Text Classification.
[2]. Jason Wei1, Kai Zou, 2019,EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks
[3].https://github.com/akkarimi/aeda_nlp/tree/master/data
[4].