Neural Processing Letters (2024) 56:121
https://doi.org/10.1007/s11063-024-11574-4

Neural Data Augmentation for Legal Overruling Task: Small Deep Learning Models vs. Large Language Models

Reshma Sheik[1] · **K. P. Siva Sundara**[2] · **S. Jaya Nirmala**[1]

Accepted: 15 February 2024

Abstract
Deep learning models produce impressive results in any natural language processing applications when given a better learning strategy and trained with large labeled datasets. However, the annotation of massive training data is far too expensive, especially in the legal domain, due to the need for trained legal professionals. Data augmentation solves the problem of learning without labeled big data. In this paper, we employ pre-trained language models and prompt engineering to generate large-scale pseudo-labeled data for the legal overruling task using 100 data samples. We train small recurrent and convolutional deep-learning models using this data and fine-tune a few other transformer models. We then evaluate the effectiveness of the models, both with and without data augmentation, using the benchmark dataset and analyze the results. We also test the performance of these models with the state-of-the-art GPT-3 model under few-shot setting. Our experimental findings demonstrate that data augmentation results in better model performance in the legal overruling task than models trained without augmentation. Furthermore, our best-performing deep learning model trained on augmented data outperforms the few-shot GPT-3 by 18% in the F1-score. Additionally, our results highlight that the small neural networks trained with augmented data achieve outcomes comparable to those of other large language models.

Keywords Deep learning · Natural language processing · Data augmentation · Legal overruling task · Transformer · Few-shot · GPT-3 · Large language models

✉ Reshma Sheik
rezmasheik@gmail.com

K. P. Siva Sundara
sivaparthi1989@gmail.com

S. Jaya Nirmala
sjaya@nitt.edu

[1] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirapalli 620015, India

[2] Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore 641013, India

Published online: 23 March 2024

1 Introduction

Deep neural network models have significantly progressed in several *Natural Language Processing* (NLP) applications. The biggest concern of deep learning is preventing overfitting and obtaining a better-generalized model. The more training data we have, the better the performance of deep learning models in any supervised learning method. Due to the wide variety of NLP tasks, majority of the task-specific datasets only contain a few thousand to a few hundred thousand manually-annotated training examples. In many deep learning techniques, the dataset influences the problem to be solved. Creating new labeled data is slow, costly, and requires trained personnel. Additionally, the labeling process is susceptible to inevitable human error. Researchers are looking for alternate ways to produce data without manual annotation. *Data augmentation* (DA) is motivated to extend the range of training examples without explicitly collecting annotated data. Data augmentation can avoid overfitting and resolve class-imbalance issues by oversampling the unrepresented label with enough training samples. DA is explored well in image processing, and it is simple to accomplish using techniques like flipping, rotating, decolorizing, enhancing the edges of images, etc. However, it is considerably more difficult with textual data because it would change the syntactic and semantic construct of the augmented data. It has recently drawn more interest in NLP because of the growth in work in low-resource areas [1]. In NLP, this could be achieved using deleting or adding words, synonym swaps, or various text generation approaches. The process of data augmentation in NLP can be addressed via Thesaurus [2], word embeddings [3–5] *back translation* [6], *text generation* [7], etc., There have been many recent advancements in pre-trained language models and its application in prompting downstream NLP tasks. These developments in generative models are revolutionizing the process of data augmentation. In these *Large Language models* (LLMs), this issue is solved by adding novel learning strategies like *few-shot learning* and *transfer learning* [8].

Transformer models like Generative Pre-trained Transformer 3 (GPT-3) [9] with hundreds of billions of parameters can achieve high-level task-specific few-shot performance comparable to state-of-the-art models. These models are based on a self-attention mechanism that allows them to process long text sequences effectively, making them ideal for tasks that require understanding and generating complex language. However, one of the major challenges of using large-scale transformer models is the massive computational resources required for training and inference, leaving a large carbon footprint and posing a challenge for academics and practitioners to use them.

Thus, we formulate interesting research questions in this direction.

Key Research Questions

- **RQ1**: Can we effectively reduce human annotation effort to train models with fair performance using augmented data?
- **RQ2**: Are student models trained on this generated data more efficient than fine-tuned large language models?
- **RQ3**: Will student models exhibit better scores than the state-of-the-art GPT-3 models tested under few-shot setting?

Motivated by this, the paper focuses on neural data augmentation using LLMs to address *Legal Overruling Task* with limited-sized datasets of 2400 samples from the Casetext law corpus.[1] The overruling task is a binary classification task, with positive examples representing overruling sentences from the law and negative ones representing non-overruling

[1] https://casetext.com/blog/a-benchmark/.

Table 1 Example sentences for overruled"1" and not overruled"0" labels from the dataset

Sentence	Label
We disapprove of the line of cases from the courts of appeal allowing attorney's fees in the case of executory process	1
For these reasons we believe that board of education v. chattin, supra, should be overruled to the extent that it conflicts with the views herein set out	1
Changes in custody or parenting time may be modified only if the moving party demonstrates that modification is justified by proper cause or because of a change of circumstances	0

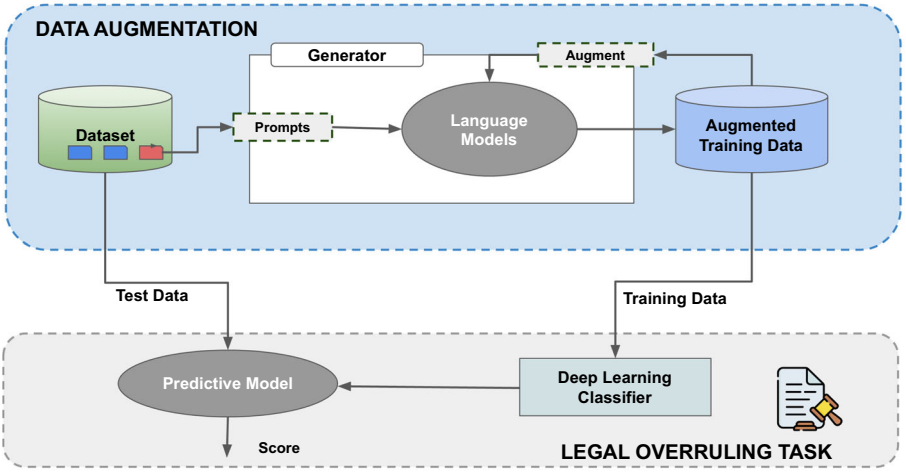

Fig. 1 Framework illustrates the process of data augmentation for legal overruling task

sentences, as shown in Table 1. An overruling sentence is a declaration that invalidates a previous case decision. The Overruling task holds great significance for lawyers as it ensures the validity of legal arguments and prevents overruled cases [10]. Identifying and labeling overruled sentences requires trained legal professionals, which is costly and time-consuming.

Our primary objective is to create synthetic data using language models with only 100 training samples. This is because deep learning models require a significant amount of data to perform well, which may not be readily available for the task of legal overruling. This is particularly relevant in real-world scenarios, where obtaining large gold-standard datasets in the legal domain can be challenging. As a result, our task is both challenging and pertinent to real-world applications. As shown in Fig. 1, we propose a data augmentation approach that utilizes a small sample of the available dataset to generate a larger set of training data via the augmentation process. We also aim to design a simple deep-learning model architecture that can attain comparable performance to large language models.

Main Contributions

The contributions of this paper can be summed up as follows:

- We generate training samples for legal overruling task using pre-trained language models (GPT-3, GPT-2 [11], Bidirectional Encoder Representation from Transformers (BERT) [12], a distilled version of BERT (DistilBERT) [13], and Robustly Optimized BERT (RoBERTa) [14]).

- Using this synthetic data, we train student models like Bidirectional Long Short Term Memory (BiLSTM), Bidirectional Gated Recurrent Unit (BiGRU), and Convolutional BiLSTM (ConvBiLSTM) and fine-tune a few transformer models like BERT, RoBERTa, and DistilBERT. Compared the performance of these models with and without augmentation
- Our work also explores the performance of large language model GPT-3 under few-shot setting for legal overruling task.
- Finally, we evaluate and compare all models based on accuracy, F1-score, model size, and inference time. We also compare with other neural text augmentation approaches to present the effectiveness of our approach. As a result, we arrived at a simple and efficient deep-learning model for the legal overruling task.

The rest of this paper is organized as follows. Section 2 presents the background study of text data augmentation, followed by neural data augmentation, and ends with current works in the legal domain. Section 3 covers the techniques used for data augmentation in legal overruling task. The model architecture and additional experimental setups for training and fine-tuning deep learning models are provided in Sect. 4. Section 5 addresses the research questions and analyses the performance of the models. Finally, Sect. 6 concludes this paper with future works.

2 Background

2.1 Text Data Augmentation

Data augmentation has recently drawn more attention in NLP due to the growth of work in low-resource domains, the emergence of new tasks, and the popularity of deep neural networks that need lots of training data. One simple and practical approach to performing text augmentation is to substitute words or phrases with their synonyms by utilizing an existing thesaurus to generate a large amount of data quickly. The paper [2] chooses a word and swaps out its synonyms based on the geometric distribution. In other methods, a related word for augmentation is identified using similarity searches such as *k-Nearest Neighbor*(k-NN), cosine similarity [3], and pre-trained traditional word embeddings like *word2vec* [15], *GloVe* [16], and *fasttext* [17]. By selecting words from the topic-word and document-topic distributions [18], used probabilistic topic models to produce more training instances. To develop a sequence generation model [19], proposed a data augmentation method that uses transformation operations given by technical experts, like a word swap. In the study, [5] finds a replacement for a target word by choosing from its context using a bi-directional deep learning architecture. The model based on *Recurrent Neural Networks* (RNN) and *Convolutional Neural Networks* (CNN) showed improved classifier performance when applied to various text classification tasks. Another work by [20] uses deep learning architectures (RNN, CNN) to perform multiple transformations on the text, including synonym replacement, random insertion, deletion, and swapping, demonstrating gains in performance when tested on five natural language processing tasks. Kafle et al. [21] created fresh samples for *visual question answering* using a template-based technique and *Long Short Term Memory* (LSTM)-based [22] strategy. The accuracy of the CNN and LSTM sentence classification models improved when interpolation-based augmentation techniques were adopted at the word and sentence embedding levels [23].

In contrast to the preceding methods, Kafle et al. [7] uses techniques to generate the entire phrase rather than just one or a few words. In [6], the concept of *back translation* is employed to translate the text information into another language and then translate it back into the original language, which makes it easier to create textual data while maintaining the context of the text data. Recent data augmentation uses counterfactual examples [24], which introduces negations or numeric alterations to flip the sample's label.

2.2 Data Augmentation Using Transformers

The success of pre-trained transformer models in many NLP applications led to conquering the data augmentation field. Earlier transfer learning approaches [25] used was to fine-tune the BERT model with an extra label-conditional constraint to enhance contextual augmentation. The examples generated by this model can also be used to perform style transfer, where they reverse the original label of the sentence, and the model outputs a new label-compatible sentence. However, they depend heavily on the amount of training data, leading to a low variance of training samples in the collection. Elsahar et al. [26] uses zero-shot learning for generating questions from knowledge graphs using an encoder-decoder neural network architecture with attention vectors. The study uses a large in-domain training dataset and a transfer learning setting for unknown knowledge base types based on existing ones. However, in-domain pretraining seems insufficient; novel pretraining and few-shot learning strategies are required in domain-specific NLP applications.

Recent improvements in text generation models, such as GPT and its later versions have sparked the creation of novel augmentation techniques that produce extra training data from original samples rather than performing local changes to the text. A similar study [27] uses GPT-2 models to generate large labeled training sets to train a BERT-based classifier for improving the performance of relation extraction tasks. Another framework [28] that makes use of a pre-trained GPT-2 model to produce label-invariant modification of the input texts to augment the existing training data for multi-label classification showed considerable gains over baseline models. In [29], two powerful transformer language models, GPT-3 and BioBERT [30] were tested in few-shot scenarios on several biomedical NLP applications. The study hypothesizes that language models may gain from domain-specific pretraining in task-specific few-shot learning. Therefore in [31], few-shot learning strategies are used for natural language generation to create a multi-domain table-to-text dataset using only 200 data samples. The same few-shot learning approach to natural language generation is used in another study [32] for data augmentation using GPT-2 models. They adopted a few seed selection strategies for extracting the learning samples. Another work language model-based data augmentation (LAMBADA) [33] uses GPT-2 to generate new sentences for the class by fine-tuning them for a particular text classification problem. The phrases generated by this method were filtered using a classifier trained on the original data to produce high-quality data. This method offers an alternative to semi-supervised techniques when unlabelled data does not exist. Another work [34] uses GPT-3 models were used to perform text augmentation via prompts and improve the robustness of pre-trained transformer models. In addition to the above generative models, in [35] demonstrated an effective method for preparing the pre-trained models for data augmentation is to prefix the class labels to text sequences. They showed that the text-to-text BART [36] model performs better than other transformer-based pre-trained models in a low-resource setting. A recent work [37] also uses fine-tuned *seq2seq* language models: T5 [38] and BART to build new samples for performance improvement in various classification tasks.

The studies and surveys [1, 39–41] indicate that data augmentation techniques can improve the performance of models trained on textual data, but the effectiveness of these techniques depends on the specific task and size of the dataset. However, it is important to note that the unique features of legal text, such as lengthy unstructured documents and legal jargon, make it challenging to apply many of the techniques commonly used in the context of normal text. As a result, additional research and development are necessary to identify effective data augmentation strategies for legal text datasets.

2.3 Data Augmentation for Legal Text

Legal documents use a separate vocabulary known as *"Legalese language"* with words with specific meanings concerning legal context. Therefore, certain words designated as protected words cannot be changed or altered. For instance, although theft and fraud are similar words in general English, they are two distinct categories of crime in legal texts [42]. These differences necessitate careful consideration while choosing an appropriate augmentation strategy. Another work [43] aims to tackle the crime prediction problem by applying three different techniques random scramble, random delete, and random insert at the sentence level, gained an 11% increase in F1-score when training data is 10,000 documents and performed significantly less when fed with 10–15 times more training data. Another technique applied to Chinese legal cases [44] performs a two-step augmentation which includes *stop word* deletion, back translation, and synonym replacement. As in [42], semantic similarity augmentation can be applied by assuming that words occurring in the same/similar contexts have similar meanings. The current solutions are applied on short texts, but it requires more runtime on long texts. The authors of [45] investigate using transformer-based decoder models to generate U.S. Court options. The trained models produced judicial opinions, which are very similar to actual opinions, and legal experts find it difficult to distinguish between them. These results point to a potential application of transformer models in the field of law. Various types of research are underway to study the impact of pre-trained models in legal domain tasks. In [46], the authors researched to explore the issues that affect pre-trained models in the legal domain.

Thus in our research, we focus on the combined way of generating samples using prompts as the initial step and further generating more samples using other neural models. We particularly focus on the legal data because a legal expert's annotation is expensive and time-consuming.

3 Methodology

3.1 Data Augmentation Process

This section presents the methodology used to augment data for the legal overruling task. To address the issue of limited data, we adopted various techniques described in the following subsections. The original overruling benchmark dataset contained only 2,400 examples, which were manually annotated by attorneys for positive overruling samples, while negative samples were randomly selected from the Casetext law corpus. However, this open dataset is insufficient to train a deep-learning network. To generate more training samples, we randomly selected a portion of this dataset, typically 100 samples, ensuring an equal split between 50 positive and 50 negative samples. Using these samples as few-shot prompts, we utilized GPT

models to generate more positive and negative samples as described in Sect. 3.1.1. Further, we employed a random subset of the data samples generated by these generative models as input for the BERT-based models (Sect. 3.1.2) and the Word2Vec model (Sect. 3.1.3) to generate more artificial samples. The algorithm 1 shows the entire neural data augmentation process for artificial sample generation by the neural network models.

Algorithm 1 Neural Data Augmentation: Artificial Samples Generation

Input: Training Data Samples: D_{train}, Language Models: G
Output: Augmented Data: D_{aug}
$D_{sub-train} \Leftarrow RandomSubset(D_{train})$
$D_{aug} \Leftarrow NULL$
Synthesize sentences D^* using generative language models G by providing k few-shot samples from $D_{sub-train}$
Remove duplicates and $NULL$ in D^*
$D_{aug} = D_{aug} \cup D^*$
$D_{sub*^1} \Leftarrow RandomSubset(D_{aug})$
$D_{sub*^2} \Leftarrow RandomSubset(D_{aug})$
$i \Leftarrow 1$
while $i < range(D_{sub*^1})$ **do**
    $d_i \Leftarrow D_{sub*^1}[i]$
    Synthesize samples d^*_{i1} and d^*_{i2} from d_i using BERT language models
    $D_{aug} = D_{aug} \cup \{d^*_{i1}, d^*_{i2}\}$
    $i \Leftarrow i + 1$
end while
$j \Leftarrow 1$
while $j < range(D_{sub*^2})$ **do**
    $d_j \Leftarrow D_{sub*^2}[j]$
    Synthesize d^*_j from d_j using Word2Vec model
    $D_{aug} = D_{aug} \cup \{d^*_j\}$
    $j \Leftarrow j + 1$
end while
return D_{aug}

3.1.1 Using Generative Pre-trained Transformer Models

Generative models are auto-regressive models developed by OpenAI, which are trained on huge data and use neural networks to predict a probable future output based on an input. To generate results, they use a unique type of neural network that processes data through multi-headed attention modules to produce results. GPT-3, 116 times larger than GPT-2 with more than 175 billion parameters, can generate human-like text and do tasks like question answering, summarization, translation, and developing codes. We used both models, which function as a generator for the augmentation process. A portion of the data, typically 100 samples, is used as few-shot examples for GPT models to generate synthetic data. Using the prompt-based generation technique, the GPT-3 model generates overruled and non-overruled sentences from the OpenAI API platform.[2] In GPT2, the text generator API from DeepAI[3] is used for the augmentation process. Later, data is cleaned to remove duplicates and missing values. Table 8 in appendix shows the sample augmented sentences generated by the models.

[2] https://beta.openai.com.

[3] https://api.deepai.org/api/text-generator.

Table 2 Legal overruling example augmented using Word2Vec model

Original sentence	Augmented sentence
The court overruled the objection and found that the evidence was significant	The court overruled the objection and found that the evidence was admissible

Table 3 Shows the number of sentences generated by each model and the time taken

Models	Number of sentences	Time taken for each sentence (s)
GPT-3	21,605	0.56
GPT-2	9269	0.192
BERT	2800	2.89
RoBERTa	2800	3.2
DistilBERT	2800	2.57
Word2vec	2100	0.144

3.1.2 Using Pre-trained BERT-Based Models

Additionally, synthetic data was produced using encoder-based transformer models, which include the BERT, RoBERTa, and DistilBERT models. NLPaug framework[4] is utilized to do the data augmentation process. A simple substitution mechanism is adopted for every model to augment data. We use 1400 samples of data generated by the generative models. For each example text, we augment two sentences using different BERT architectures. Table 9 in appendix shows the data augmented using BERT Architectures.

3.1.3 Using Pre-trained Word2Vec Model

For the augmentation process using the Word2Vec language model, we used a random sample of 2100 data generated by the generative models. We started the augmentation process with the simple synonym replacement strategy based on contextual word embedding (Word2Vec) [47]. Table 2 shows the sample legal sentence generated using the Word2Vec neural model. We used a single-word replacement strategy for the augmentation process using the NLPaug framework at the word level.

We combined all data produced by the neural models into a single file to proceed with the binary classification of the legal overruling task. Table 3 shows the number of artificial samples produced by each model and the time taken to augment each sentence. We obtained a highly balanced data set of 20,854 overruled samples and 20,520 non-overruled examples, resulting in a total of 41,374 instances. The word distribution of the complete augmented data is shown in Fig. 2.

3.2 Model Architecture

To evaluate the performance of the legal overruling task across different deep learning models, a diverse set of student models was trained using artificial data. The architectural details of

[4] https://github.com/GEM-benchmark/NL-Augmenter.

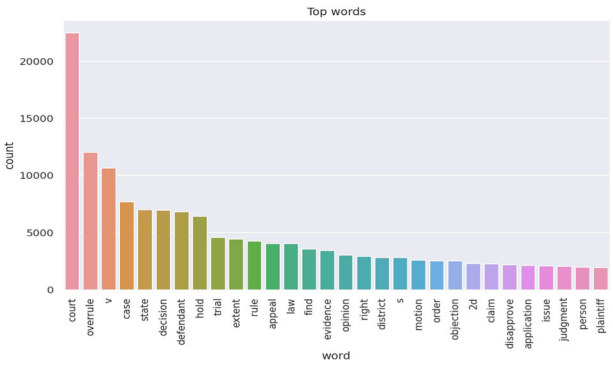

Fig. 2 Word frequency distribution of the augmented data

these models were detailed in Sect. 3.2.1. We also fine-tuned BERT-based large language models and presented them in 3.2.2.

3.2.1 Deep Learning Models

This section discusses the architecture of the various student models used for the legal overruling task.

Bidirectional LSTM: Bidirectional long-short-term memory (BiLSTM) is the deep learning technique to process sequence data in both directions, forward and backward. In our experiment, the BiLSTM model was defined to include a 300-dimensional embedding layer. It was then followed by three layers of BiLSTM with sizes of 128, 64, and 32 each, followed by a dropout layer. The dense layer for classification was equipped with one hidden layer of size 32 with ReLU activation [48], later fed to the output layer with sigmoid activation.

Convolutional BiLSTM: This architecture uses a convolutional layer as a feature extractor layer in the BiLSTM framework. The architecture starts with 500 dimensional embedding layer. This was followed by a convolutional layer with a kernel size of three and 32 filters, subsequently followed by a max-pooling layer. Then in sequence, two layers of BiLSTM of 128 and 64 hidden sizes followed by the dropout layer. The final output layer is a fully connected dense layer with sigmoid activation.

Bidirectional GRU: The architecture in which our Bidirectional Gated Recurrent Unit (BiGRU) was defined is similar to that of BiLSTM. The architecture starts with an embedding layer of size 500, the subsequent components comprised three BiGRU layers of 256, 128, and 64 sizes. In sequence, each BIGRU layer is followed by a dropout layer. The final layer is a fully connected dense layer with a hidden layercomprising 32 neurons activated by ReLu and an output layer with sigmoid activation.

3.2.2 Large Language Models

We fine-tuned the following bert-base language models as these models were used for the data augmentation process in Sect. 3.1.2.

BERT: BERT is a model pre-trained on a large corpus of English data in a self-supervised fashion. This implies that an automatic method to create inputs and labels from those texts was pre-trained on the raw texts without human labeling effort [12].

RoBERTa: RoBERTa [14] is an optimized version of BERT trained on a much larger dataset with an effective training procedure.

DistilBERT: DistilBERT is a compact, fast, and cheap transformer model trained by a distilling BERT base. It runs 60% faster and uses 40% fewer parameters than bert-base-uncased while retaining over 95% of BERT's performance on language understanding tasks [13].

Fine-tuning LLMs, like BERT, RoBERTa, and DistilBERT, requires the input data to be in a specific format, so the data must be processed to convert it into this format. This involves tokenizing the text, creating attention masks, and converting the labels into numerical values. The default tokenizer class, *AutoTokenizer*, sourced from the Hugging Face Transformer library,[5] is used for data processing and subsequent fine-tuning of the models.

4 Experimental Details

This section provides an overview of our experimental setup for data augmentation, training parameters for model classification, and few-shot parameter tuning. The last subsection outlines the evaluation setups used in our work to assess the performance of the trained classifiers. The whole experiment was carried out in the Google colaboratory notebook pro-environment and implemented using Python.

4.1 Setup for Data Augmentation

Data augmentation with generative models comes with certain limitations and difficulties. One of the main difficulties is the usage policies and usage limits set by OpenAI. These policies are designed to ensure the model's responsible usage and prevent misuse. For example, there are limits on the number of API calls that can be made per month, making it challenging for researchers to collect large amounts of data for their experiments. Additionally, GPT-3 is restricted for certain types of usage, such as using the model for illegal or unethical purposes. This can limit the scope of data samples generated with the model. For data generation using GPT-3, we used the completion function from the OpenAI library and the davinci-002 model.[6] The model will produce the best results for applications that require a high level of understanding of the content, such as creative content generation. More computing resources are required to support these enhanced capabilities. We have set the temperature to 0.45. The temperature parameter controls the randomness in the model's behavior. Lowering the value results in fewer random completions. As prompt engineering, the model uses a short description of the task and an example sentence to generate new synthetic data. We used DeepAI's text generator API to access the GPT-2 model to produce synthetic data. We adopted the default setting for model parameters in GPT-2, and text was generated using the prompt with examples as input. We sourced the BERT-based models for text augmentation from the Hugging Face pre-trained transformer library.

[5] https://huggingface.co/models.

[6] https://beta.openai.com/docs/models/gpt-3.

Table 4 Models and its attributes for training

Model	Model size	#Parameters	#Epochs	Training time/ epoch
BiLSTM	93 mb	91,73,665	17	58
BiGRU	121 mb	1,38,86,865	12	55
ConvBiLSTM	76 mb	80,71,433	15	19
BERT	367 mb	10,83,11,810	3	5520
RoBERT	475 mb	12,46,47,170	3	5586
DistilBERT	255 mb	6,69,55,010	3	2820

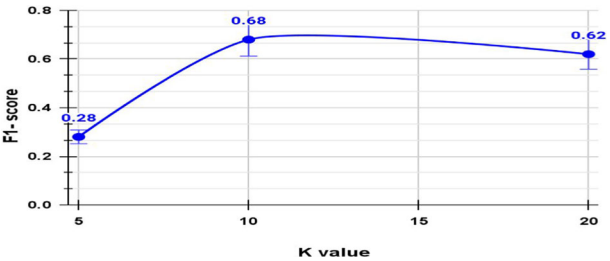

Fig. 3 The optimal K value, which represents the number of few-shot samples fed to the model to be determined to test the best performance of the GPT-3 model

4.2 Training Parameters for Model Classification

For training the deep neural network framework (BiLSTM, BiGRU, and ConvBiLSTM), the optimizer was adam [49] with binary cross entropy as the loss function. A dropout of 0.2 was used after every hidden layer. These models underwent 20 epochs of training and used early stopping based on validation accuracy. Hyperparameter tuning was performed using a Hyperband tuner from the Keras library[7] to determine the appropriate learning rate and the optimal number of neurons for each layer. We have adopted a learning rate of 0.001 from choices [1e-2, 1e-3, 1e-4, 1e-5], and neurons were selected in the range of 32 to 512 with a step size of 32. For the transformer models, the base version of pre-trained LLMs from the hugging face library was loaded for the sequence classification task. The models were then fine-tuned on the augmented data for three epochs with a learning rate of 0.00001. Table 4 shows the model size and the optimal number of epochs with training time.

4.3 Parameter Setting for Few-Shot GPT-3

To evaluate the performance of the GPT-3 model under the few-shot setting, multiple values for K (which specifies the number of examples fed to the model as prompts before testing) were experimented with, and the F1-score was recorded, revealed in Fig. 3 that the best outcome was achieved when K was equal to 10, with the F1-score decreasing when K was set to 15.

Thus, the GPT-3 model was tested on the original overruling dataset of 2300 samples, with a K value of 10.

[7] https://keras.io/api/keras_tuner/tuners/hyperband/.

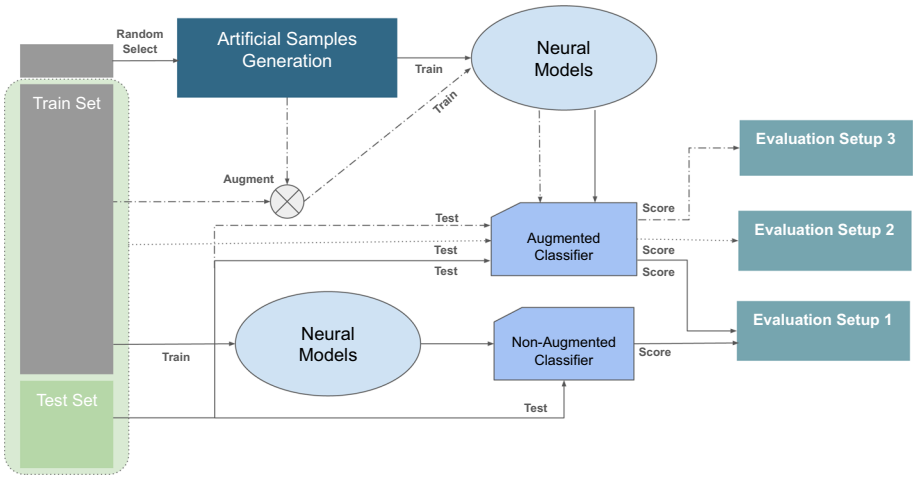

Fig. 4 The various evaluation strategies used for the augmented classifiers and non-augmented neural classifiers

4.4 Evaluation Setup

To assess the performance of trained classifiers, we use three evaluation setups. In the first approach, we compare the performance of the augmented and non-augmented classifiers. To establish a strong baseline for comparison, the non-augmented classifier undergoes training on the original 80% of human-labeled data (train-set), while the augmented classifier is trained on artificially generated samples using a random subset of 100 samples from the train-set. Both these models were tested on the remaining 20% test-set for comparison. In the second setup, the augmented classifier trained with artificial samples was tested on the remaining gold-standard data to to evaluate the performance of various deep learning models, including the few-shot GPT-3. The third evaluation setup involves training neural models by incorporating the artificially generated samples into the original train-set. This allows us to compare their performance against various neural data augmentation techniques serving as baselines. Figure 4 outlines the various evaluation strategies used for both classifiers.

5 Results and Discussion

The results of our experiments corresponding to evaluation setup 1, 2, and 3 are presented in Sects. 5.1, 5.2, and 5.3, respectively. Finally, we address and discuss the key research questions in Sect. 5.4.

5.1 Performance of the Models With and Without Augmentation

In this study, we compare the classifier's performance with and without augmentation, based on the existing dataset comprising 2,400 samples. Training the non-augmented classifiers involved allocating 80% of the dataset, while the remaining 20% was reserved for testing both augmented and non-augmented models for comparative analysis. We also used a statistical significance test to evaluate the difference in performance.

Table 5 Models classification performance with and without augmentation

Model	Without augmentation		With augmentation	
	Accuracy	F1-score	Accuracy	F1-score
BiLSTM	0.75	0.75	0.84	0.87
BiGRU	0.79	0.8	0.83	0.86
ConvBiLSTM	0.8	0.79	0.86	0.85
BERT	0.81	0.78	0.87	0.82
RoBERTa	0.86	0.81	0.92	0.88
DistilBERT	0.77	0.73	0.83	0.81

Table 5 shows the performance of the models with and without augmentation based on accuracy and F1-score. It is observed that the RoBERTa models had a better score in accuracy and F1-score than the other deep learning models. Among the student models, BiLSTM had a relative increase in accuracy and F1-score compared to other models trained with augmented data. The Receiver Operating Characteristic (ROC) curve is used to further analyze classifier performance. Figure 5 shows that the area under the ROC curve is larger for models trained with augmentation than without augmentation.

The *paired t-test* [50], which compares the means of two groups, was used as the statistical test to assess further the impact of augmentation on the classifier's performance. It is used to establish whether or not the mean difference between the pairs of observation is zero. It is commonly used to determine whether two groups vary or whether a process or treatment significantly affects the population of interest. We use this test to measure whether the augmented training samples significantly improve the classifier's performance compared to the non-augmented classifier. We set the hypothesis as follows:-

Null Hypothesis H_o: The performance of classifiers does not increase due to augmented data.

Alternate Hypothesis H_a: The performance of classifiers has increased due to augmented data.

As a threshold, we use α as 0.05. The following equation 1 represents the test statistic t.

$$t = \frac{\bar{X} - \mu}{s/\sqrt{n}} \tag{1}$$

where μ denotes the hypothesized population mean with $\bar{X}$ as sample mean and s as standard deviation with n denotes the number of classifier models. On evaluation, it results in a test statistic value of -9.428473 and obtains a $p - value$ for one_tailed_test as 0.000113, a value that is significantly lower than α value. Thus t value falls into the critical region, thereby rejecting the null hypothesis and concluding that augmented techniques improve classifier performance compared to non-augmented classifiers.

5.2 Comparison Between Models

In this research, we performed extensive experiments, studying different configurations to compare and contrast the deep learning models. This study used the following criteria to determine the best model.

- Relatively high F1-score
- Relatively small model size

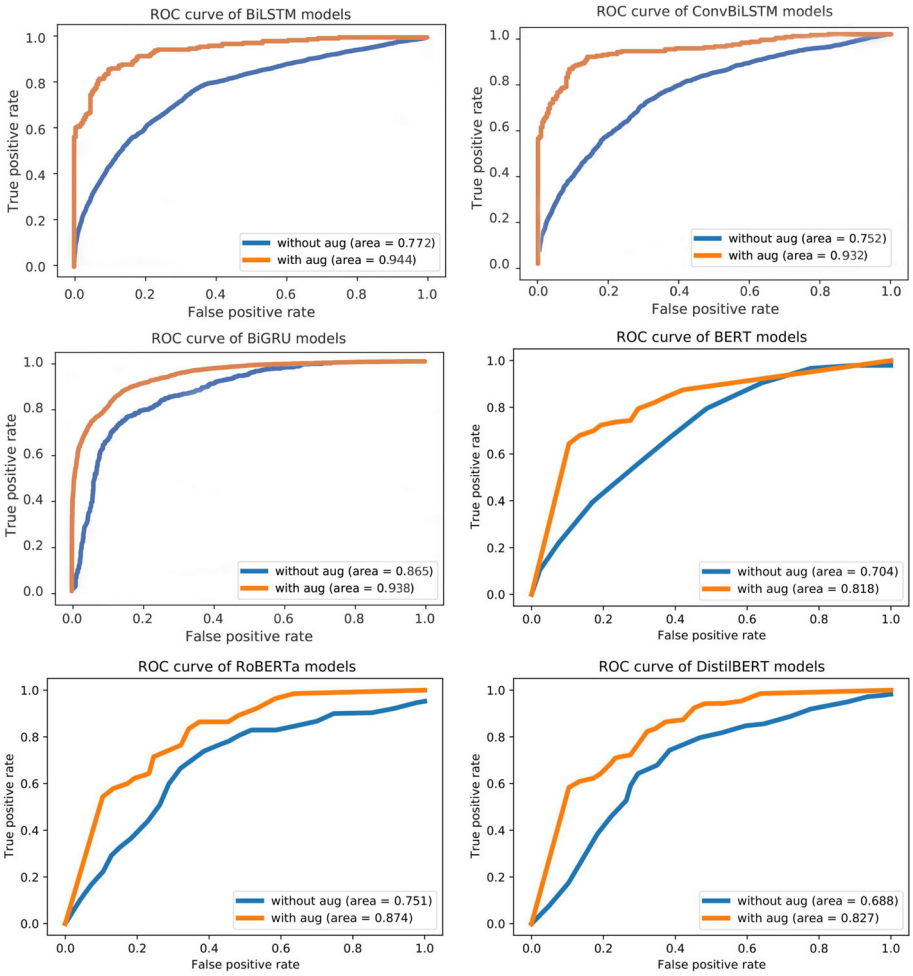

Fig. 5 ROC curves for the models trained with and without augmentation

- Relatively low inference time

Our study assessed the efficacy of several deep learning models trained on the artificial samples for the legal overruling task by evaluating their performance on 2300 unseen test samples. We compared the models based on three key metrics: F1-score, model size, and inference time. While GPT-3 has been shown to outperform many state-of-the-art models in natural language processing applications, it requires significant computational resources to achieve optimal results. To establish a baseline for comparison, we analyzed the performance of GPT-3 under a ten-shot setting. Table 6 shows the metrics for the classification performance of all the deep learning models. Our findings indicate that the ten-shot GPT-3 model achieves an F1-score of 0.69 for the legal overruling task. However, the fine-tuned encoder-based transformer models performed better than the few-shot GPT-3 model. Specifically, RoBERTa base had the highest F1-score of 0.9, while the BiGRU student model led with an accuracy of 0.96 highlighted in bold. The table also presents findings related to the inference time

Table 6 Model classification performance

Model	Accuracy	Precision	Recall	F1-score	Inference time (s)
BiLSTM	0.9	0.85	0.85	0.85	8
BiGRU	**0.96**	0.87	0.87	0.87	20
ConvBiLSTM	0.87	0.83	0.84	0.83	11
BERT	0.92	0.88	0.86	0.87	110
RoBERTa	0.93	0.9	0.91	**0.9**	146
DistilBERT	0.86	0.89	0.87	0.88	70
GPT-3[1]	0.74	0.86	0.57	0.69	1029

[1] The GPT-3 model is tested under few-shot scenario
Bold values indicates the metrics values of best performing models

Table 7 Models classification performance with and without augmentation

Data augmentation approach/ Model for classification	BiGRU		RoBERTa	
	Accuracy	F1-score	Accuracy	F1-Score
Using samples generated by Word2Vec	0.94	0.94	0.95	0.96
Using samples generated by BERT	0.93	0.94	0.96	0.97
Using samples generated by GPT2	0.92	0.93	0.93	0.93
Using samples generated by GPT3	0.90	0.90	0.92	0.91
Proposed approach	**0.96**	**0.96**	**0.97**	**0.97**

Bold values indicates the metrics values of best performing models

in seconds for each individual model. It is observed that the inference time of fine-tuned transformer models was several times higher than the student deep-learning models.

5.3 Comparison to Other Neural Data Augmentation Techniques

In this section, we individually compare the various neural text data augmentation techniques with our proposed hybrid approach. The samples are generated using neural models ranging from the traditional Word2Vec model [15] to transformer models[25, 35], including generative pre-trained models such as GPT-2 [28, 31] and GPT-3 [34]. The samples produced by these techniques were augmented to the existing training data for classification. We have performed this classification deploying the best-performed small deep learning model, BiGRU, and the best-performed LLM, RoBERTA, based on the result of the experiment conducted in Sect. 5.2. Table 7 shows that our proposed approach leads in accuracy and F1-score for both models.

5.4 Discussion

GPT-3 is a large and complex model requiring significant computational resources, making it difficult for researchers with limited computational resources to experiment with the model. Further, a significant issue with text generation models is the possibility of noise generation, which lowers classification model performance instead of improving it [51]. From Table 6, the RoBERTa base model produces F1-scores that outperform those of the GPT-3 model,

indicating that the RoBERTa model is a more efficient option for the legal overruling task. In fact, it produces results that are the best among all models evaluated. However, it is noteworthy that despite the RoBERTa base model being smaller than the GPT-3 model and having fewer trainable parameters, the student models exhibit similar performance levels. More parameters aid the RoBERTa model in performing better, but its larger size results in huge training time. On the other hand, the student models' training time is merely a fraction of the fine-tuning time of transformer models, making them a more practical and efficient solution.

After analyzing all the information presented, the BiGRU model trained with augmented data stands out as the best option for the legal overruling task. Despite being one of the smallest models, the BiGRU model requires less training time and has fewer trainable parameters while still exhibiting high accuracy. In fact, it outperforms the GPT-3 model under few-shot conditions and competes well with other large language models in terms of the F1 score. Simple RNNs like BiGRU are preferred to RoBERTa because of their simpler architecture, smaller size, and faster runtime. Since the positive and negative overruling examples are taken from the casetext law corpus and the terminology used in the samples is specific to the legal domain, overruling has moderate to high domain specificity [52]. LLMs like GPT-3 are trained on a vast English corpus different from the legal corpora, which could be the reason for the lower performance of GPT-3 under few-shot setting. These limitations should be considered when planning and conducting research with GPT-3. Overall, the evidence suggests that small deep learning models trained on domain-specific data may offer a more efficient and effective approach to domain-specific NLP tasks like legal overruling. While larger pre-trained models like GPT-3 and RoBERTa are impressive in their ability to handle a wide range of tasks, they may not always be the best option for specific domain-specific tasks. By leveraging domain-specific data and designing models that are optimized for specific tasks, researchers can potentially achieve better accuracy and faster inference times while using fewer computational resources.

Answers to Research Questions

RQ1: *Can we effectively reduce human annotation effort to train models with fair performance using augmented data ?* Yes, all the models trained with the augmented data performed better in the test data (Table 5) compared to the results achieved with non-augmented data. Further, the outcomes of the paired t-test and ROC curves (Fig. 5) strengthened the same finding. Hence, it significantly reduces expensive human annotation in training models for the legal overruling task, thereby improving the generalization power of deep learning models.

RQ2: *Are student models trained on this generated data more efficient than fine-tuned large language models?* Yes, our results indicate that it is indeed possible to achieve comparable performance to large fine-tuned language models using small deep-learning models trained with augmented data, even with limited computing power and memory resources, as demonstrated in Tables 6 and 7.

RQ3: *Will student models exhibit better scores than the state-of-the-art GPT-3 models tested under few-shot setting?* Yes, according to our findings on the legal overruling task, our best-performed deep learning model outperforms GPT-3 models tested in few-shot settings by over 18%. Additionally, it is much faster to run these student models, which leads to a better inference score that is way higher than GPT-3 models (Table 6). Thus dealing with state-of-the-art models suffers from economic costs, latency in usage, and high memory footprints.

6 Conclusions and Future Work

The research utilizes neural models for text data augmentation to enhance performance in low-resourced tasks. We augmented 41,374 data samples with 100 training examples for the legal overruling task using neural models. Our study proposed three simple neural network architectures: BiLSTM, BiGRU, and ConvBiLSTM. The result showed that data augmentation strengthens the classification process leading to robust classifiers. Our experiments also use the state-of-the-art GPT-3 models tested under a few-shot scenario for the legal overruling task. As GPT-3 was trained on a generic domain, it failed to produce results with high accuracy for the domain-specific task. Instead, small deep-learning neural networks trained on a large amount of augmented data have results that outperform GPT-3 and are nearly equal to other fine-tuned transformer models. After testing all models, results show that the BiGRU model being smaller in size and with less inference time, outperforms the GPT-3 model and competes with fine-tuned RoBERTa model, reducing the carbon footprint caused by these LLMs. As a result, our work demonstrated the deduction of expensive manual data annotation through data augmentation for such domain-specific tasks, which has great potential for facilitating the training of small, robust deep learning models. Furthermore, despite their smaller size, the student models still deliver nearly equal performance to the larger models, making them a compelling option for researchers looking to develop domain-specific NLP models with limited computational resources.

Handling protected words in legal texts is another challenge in augmentation, which can be addressed using a knowledge base of such words. Since pre-training using a general corpus is not viable in the legal domain, data augmentation based on domain knowledge, such as substitution based on a list of words that can be replaced, may be effective. In the future, we plan to use domain-specific transformer models in the augmentation process, which can produce more realistic samples. Additionally, the class imbalance issue can be resolved by using a few seed selection strategies for extracting different samples from the training data for the augmentation process. Domain specialists can make the process more efficient; therefore, in the future, legal NLP research could consider modeling active learning approaches to take these data variations into account.

Data availability and ethical concerns: The Overruling task dataset used in this study was provided by Casetext, a company specializing in legal research, and is open for research. The augmented version of this dataset generated as part of this work is made available upon request to the authors. The code used in this study for augmentation and modeling was developed using open-source tools and libraries, and the authors are willing to make the code accessible upon request. This would enable other researchers to reproduce the experiments and build upon the research findings, potentially leading to new insights.

Declarations

Conflict of interest All authors certify that they have no involvement in any firm or entity with any financial or non-financial interest in the materials covered in this document.

Appendix A Results of Data Augmentation

Here, we present the results of the augmentation process by using large language models. Table 8 shows the sample sentences by GPT models. Table 9 presents the sentences augmented by the BERT architectures.

Table 8 Augmented sentences using generative models

Augmented sentence	Label
The fact that the defendant had access to the victim's house at the time of the crime was of no effect in the decision, and insofar as the decision is inconsistent therewith, it must be overruled	1
The court's decision in Langley is hereby overruled to the extent that it suggests that a defendant's demeanor in court may give rise to a reasonable inference of guilt	1
The court may not review the Attorney General's decision to deny a request for a stay of deportation	0
The admissions office also undertakes regular "reliability analyses" to ensure that its admissions decisions are made in accordance with the law	0

Table 9 Augmented examples generated by the Encoder-based transformer models

Model	Original sentence	Augmented sentence	Label
BERT	to the extent that this dictum is inconsistent with our holding here, we expressly overrule it	on the fact that his dictum is inconsistent with any authority here, we therefore overrule it	1
	but detective kabler's report states that while detective baker conducted the initial interview with defendant, both detectives ultimately interviewed him together	but steve lyons'official report states that while detective robinson conducted that initial interview against defendant, said detectives ultimately interviewed prosecutors together	0
RoBERTa	to the extent that these decisions are inconsistent with the views hereinafter expressed, they are disapproved	to cause appearance that these decisions constitute unlawful or the views hereinafter expressed, and be disapproved	1
	a subsequent search of the vehicle revealed the presence of an additional syringe that had been hidden inside a purse located on the passenger side of the vehicle	a physical search of third vehicle revealed the presence of the additional syringe that may been hidden inside two compartments located around the passenger side inside each vehicle	0
DistilBERT	as discussed elsewhere, the absence of that language in the mlssa supports overruling boutte	as observed elsewhere, that absence of formal language defining the definition supports overruling definitions	1
	bradley fails to show that a rational jury could not have found that the evidence, viewed in a light most favorable to the prosecution, supports her conviction beyond reasonable doubt	bradley sought to show whether a rational jury could not truly proved that the evidence, viewed in a light most favorable in the defendants, denied false conviction beyond false suspicion	0

References

1. Feng SY, Gangal V, Wei J, Chandar S, Vosoughi S, Mitamura T, Hovy E (2021) A survey of data augmentation approaches for nlp. In: Findings of the association for computational linguistics: ACL-IJCNLP 2021, pp 968–988
2. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. Adv Neural Inf Process Syst 28
3. Wang WY, Yang D (2015) That's so annoying!!!: A lexical and frame-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. Association for Computational Linguistics, Lisbon, Portugal (2015)
4. Fadaee M, Bisazza A, Monz C (2017) Data augmentation for low-resource neural machine translation. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 2, pp 567–573. Association for Computational Linguistics, Vancouver, Canada
5. Kobayashi S (2018) Contextual augmentation: Data augmentation by words with paradigmatic relations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 2, pp 452–457. Association for Computational Linguistics, New Orleans, Louisiana
6. Sennrich R, Haddow B, Birch A (2016) Improving neural machine translation models with monolingual data. In: Proceedings of the 54th annual meeting of the association for computational linguistics, vol 1, pp 86–96. Association for Computational Linguistics, Berlin, Germany
7. Kafle K, Yousefhussien M, Kanan C (2017) Data augmentation for visual question answering. In: Proceedings of the 10th international conference on natural language generation, pp 198–202. Association for Computational Linguistics, Santiago de Compostela, Spain
8. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big Data 3(1):1–40
9. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
10. Zheng L, Guha N, Anderson BR, Henderson P, Ho DE (2021) When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In: Proceedings of the eighteenth international conference on artificial intelligence and law, pp 159–168
11. Radford A, Wu J, Amodei D, Amodei D, Clark J, Brundage M, Sutskever I (2019) Better language models and their implications. OpenAI blog 1:2
12. Kenton JDMWC, Toutanova LK (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp 4171–4186
13. Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108
14. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
15. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
16. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
17. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
18. Chen E, Lin Y, Xiong H, Luo Q, Ma H (2011) Exploiting probabilistic topic models to improve text categorization under class imbalance. Inf Process Manag 47(2):202–214
19. Ratner AJ, De Sa CM, Wu S, Selsam D, Ré C (2016) Data programming: Creating large training sets, quickly. Adv Neural Inf Process Syst 29
20. Wei J, Zou K (2019) EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 6382–6388. Association for Computational Linguistics, Hong Kong, China (2019)
21. Kafle K, Yousefhussien M, Kanan C (2017) Data augmentation for visual question answering. In: Proceedings of the 10th international conference on natural language generation, pp 198–202
22. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
23. Guo H, Mao Y, Zhang R (2019) Augmenting data with mixup for sentence classification: an empirical study. arXiv
24. Kaushik D, Hovy E, Lipton ZC (2019) Learning the difference that makes a difference with counterfactually-augmented data. arXiv preprint arXiv:1909.12434 (2019)

25. Wu X, Lv S, Zang L, Han J, Hu S (2019) Conditional bert contextual augmentation. In: International conference on computational science, pp 84–95. Springer
26. Elsahar H, Gravier C, Laforest F (2018) Zero-shot question generation from knowledge graphs for unseen predicates and entity types. arXiv preprint arXiv:1802.06842
27. Papanikolaou Y, Pierleoni A (2020) Dare: data augmented relation extraction with gpt-2. arXiv preprint arXiv:2004.13845
28. Zhang D, Li T, Zhang H, Yin B (2020) On data augmentation for extreme multi-label classification. arXiv preprint arXiv:2009.10778
29. Moradi M, Blagec K, Haberl F, Samwald M (2021) Gpt-3 models are poor few-shot learners in the biomedical domain. arXiv preprint arXiv:2109.02555
30. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36(4):1234–1240
31. Chen Z, Eavani H, Chen W, Liu Y, Wang WY (2019) Few-shot nlg with pre-trained language model. arXiv preprint arXiv:1904.09521
32. Edwards A, Ushio A, Camacho-Collados J, de Ribaupierre H, Preece A (2021) Guiding generative language models for data augmentation in few-shot text classification. arXiv preprint arXiv:2111.09064
33. Anaby-Tavor A, Carmeli B, Goldbraich E, Kantor A, Kour G, Shlomov S, Tepper N, Zwerdling N (2020) Do not have enough data? Deep learning to the rescue! In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 7383–7390
34. Yoo KM, Park D, Kang J, Lee SW, Park W (2021) Gpt3mix: Leveraging large-scale language models for text augmentation. In: Findings of the association for computational linguistics: EMNLP 2021, pp 2225–2239
35. Kumar V, Choudhary A, Cho E (2020) Data augmentation using pre-trained transformer models. In: Proceedings of the 2nd workshop on life-long learning for spoken language systems, pp 18–26. Association for Computational Linguistics, Suzhou, China (2020)
36. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461
37. Chen Y, Liu Y (2022) Rethinking data augmentation in text-to-text paradigm. In: Proceedings of the 29th international conference on computational linguistics, pp 1157–1162. International Committee on Computational Linguistics, Gyeongju, Republic of Korea
38. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer
39. Okimura I, Reid M, Kawano M, Matsuo Y (2022) On the impact of data augmentation on downstream performance in natural language processing. In: Proceedings of the third workshop on insights from negative results in NLP, pp 88–93
40. Shorten C, Khoshgoftaar TM, Furht B (2021) Text data augmentation for deep learning. J Big Data 8(1):1–34
41. Bayer M, Kaufhold MA, Reuter C (2021) A survey on data augmentation for text classification. ACM Comput Surv (2021)
42. Csányi G, Orosz T (2022) Comparison of data augmentation methods for legal document classification. Acta Technica Jaurinensis 15(1):15–21
43. Yan G, Li Y, Zhang S, Chen Z (2019) Data augmentation for deep learning of judgment documents. In: International conference on intelligent science and big data engineering, Springer, pp 232–242
44. Guo Z, Liu J, He T, Li Z, Zhangzhu P (2020) Taujud: test augmentation of machine learning in judicial documents. In: Proceedings of the 29th ACM SIGSOFT international symposium on software testing and analysis, pp 549–552
45. Peric L, Mijic S, Stammbach D, Ash E (2020) Legal language modeling with transformers. In: Proceedings of the fourth workshop on automated semantic analysis of information in legal text (ASAIL 2020) held online in conjunction with te 33rd international conference on legal knowledge and information systems (JURIX 2020) December 9, 2020, vol 2764 (2020). CEUR-WS
46. Nguyen HT, Nguyen LM (2021) Sublanguage: A serious issue affects pretrained models in legal domain. arXiv preprint arXiv:2104.07782
47. Bonthu S, Dayal A, Lakshmi M, Rama Sree S (2022) Effective text augmentation strategy for nlp models. In: Proceedings of third international conference on sustainable computing, pp 521–531. Springer
48. Agarap AF (2018) Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375
49. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Bengio Y, LeCun Y (eds) 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
50. Hsu H, Lachenbruch PA (2014) Paired t test. Wiley StatsRef: statistics reference online

51. Yang Y, Malaviya C, Fernandez J, Swayamdipta S, Le Bras R, Wang JP, Bhagavatula C, Choi Y, Downey D (2020) Generative data augmentation for commonsense reasoning. In: Findings of the association for computational linguistics: EMNLP 2020, pp 1008–1025. Association for Computational Linguistics
52. Chalkidis I, Jana A, Hartung D, Bommarito M, Androutsopoulos I, Katz DM, Aletras N (2021) Lexglue: abenchmark dataset for legal language understanding in english. arXiv preprint arXiv:2110.00976