A Federated Learning approach for text classification using NLP

No Author Given

No Institute Given

Abstract. Text classification is important in many aspects of NLP, such as word semantic categorization, emotion analysis, question answering, and conversation management. Law, health, and marketing are just a few of the professions that have made use of it throughout the last century. We focused on emotion analysis in this research, which includes categories like happiness, sadness, and more. We investigated the precision of three alternative models for assessing the emotional tone of written material. Deep learning models like GRU (Gated Recurrent Unit) and CNN (Convolutional Neural Network) are used in conjunction with the Bi-LSTM (Bidirectional LSTM) model. These three major deep learning architectures have been extensively studied for classification applications. Finally, the model with the greatest accuracy on the dataset was trained using Federated Learning. Using the FL approach, more data may be collected, eliminating data gaps and ensuring data security. The focus of this research is to increase the model's accuracy by collaborating with FL approaches while maintaining data confidentiality.

Keywords: : NLP · Federated · Deep learning · CNN · GRU · Bi-LSTM · ML

1 Introduction

Text may be categorized using a variety of approaches, such as the number of words in a paragraph or the genre of the text. It is possible to employ text classification for a variety of NLP tasks, such as sentiment analysis, question answering, and subject identification. A text classification issue is often broken down into four stages: dimension reduction, text preparation, text classification, and text assessment. Text categorization may be useful in natural language processing (NLP) tasks including sentiment analysis, question answering, and topic labeling. Text classification problems often include four-step processes: dimension reduction, text preparation, classification, and evaluation. However, the main issue is uploading or sharing text data in order to improve model performance, which has previously been achieved utilizing deep learning models. It's not always practicable because of the various privacy concerns of the customers. It's done with the help of a sophisticated deep learning model. Despite the fact that deep learning models have already attained the highest levels of performance in text categorization, uploading and sharing text data is still a significant barrier. This method is implemented via the usage of a pre-trained deep learning

model. One way to keep text classification private is to use federated learning. With the introduction of Google's federated learning in 2016, service providers may benefit from models that have access to private data without needing to access such data or aggregate all the information into a single dataset. If many devices were trained together, the model would be distributed to each device for additional refinement and improvement. [1] There are numerous clients working together to solve a machine learning problem under the direction of one central server or service provider in the Federated Learning model. Training machines via federated learning is a decentralized method. No third party receives any of the raw data from any of the clients. The framework's database was used to train local models. When the central server has acquired gradients from all platforms and has completed collecting them, a global model update is generally done. Once this procedure is complete, a new set of parameters is transmitted to each platform for the next round of modeling training.

2 Related Work

Liu and Miller[2] presented the federated pre-training and fine-tuning of the BERT model using clinical data. They are the first ones who used the federated settings for a large transformer model. In a federated manner, they trained the clinical corpus with both pre-training and fine-tuning methods while using multiple silos of data. Even though original BERT model was trained on Books and Wikipedia but they did it on clinical data using MIMIC-III discharge summary using a federated manner and for that they conducted 6 different experiments.

The research [3] proposed a realistic technique for dealing with out-of-domain issues utilizing continuous speech-based models like wake word detectors. Precision is more critical since the model may be utilized at any time. An embedded wake word detector is used to investigate the usage of federated learning. They offer a stochastic gradient descent optimization method that uses both local and global per-coordinate gradient scaling to increase convergence. How to train a wake word detector with entirely decentralized user data will be the subject of the first section of this article. Next, the federated optimization technique and how to replace its global averaging with an Adam-inspired adaptive averaging algorithm will be covered.

Federated learning employs client devices to train a Neural Network model, which is subsequently delivered back to the machine learning model's owner, the server. The server compiles these models into a single global model and delivers it back. We repeat this until the global model reaches our desired accuracy. Any transaction and model data in FL is stored on the server. The records and model stored on the server will be lost or destroyed if it is damaged or malfunctioned. If this happens, the model will need to be retrained. [6]

FedNer [7] is a Framework where the server usually coordinates multiple clients for the purpose of local model updating and global model sharing. Basically, here the clients are different medical platforms, and they used to train their local models with the data that is stored privately. FedNer is also implemented in different places like Dataset and Experimental Settings, Experimental Results, Influence of Training Data Size, Model Decomposition Strategy, Influence of Overlapped Entity Number and Generalization of FedNer Framework.

Healthcare access [8] is proving to be a significant difficulty. Through social media, users from all around the world may share their views and ideas. The categorization job is the most important aspect of this research. Text data may come from a variety of places, including emails, chats, site data, customer evaluations, and feedback. The accuracy of the algorithms employed in DL is used to assess the performance for categorizing the text. The process of text classification include categorizing news, assessing the text based on user opinion, and classifying the content. For NLP tasks, the standard RNN model failed miserably. Deep learning in combination with natural language processing (NLP) offers great potential for text classification challenges.

3 Input Data

3.1 Data-Set

We have a fairly well-labeled dataset that comprises around 21450 unique data points of multi-class emotions. Our dataset contains more than simply sentiment classification; there is more to a sentence's sentiment than just positive, negative, and neutral sentiment. We were trying to figure out what was going on in the mind of the person who wrote the words. Sadness , rage, surprise, joy, love and fear are just a few of the emotions represented in this multi-class data collection. We chose to create a csv file from all of the text files since arranging the data from a text file is very time-consuming and difficult. The csv file has 2 columns and they are text and emotions. The Emotions column has a number of different categories, ranging from happy to sadness, to love and fear.

Text	Emotion
i didn't feel humiliated	sadness
i can go from feeling so hopeless to so damned	sadness
i'm grabbing a minute to post i feel greedy wrong	anger
I am ever feeling nostalgic about the fireplace	love
i am feeling grouch	anger

3.2 Data Pre-processing

We have previously processed the data before delving into the model's work. This data processing will allow the models to operate more effectively and with

fewer mistakes. To begin, we eliminated the emoji symbols from each sentence in the dataset so that the models could only deal with text. After that, we removed any URLs from the texts. We don't require the URL in our dataset since it does not indicate any kind of emotion. Unnecessary punctuation marks may appear in the text. Unnecessary punctuation may lead us to wrong evaluation , so we eliminated all unnecessary punctuation. We divided the data into three sets: the training dataset (which contains about 70 percent of the whole dataset), validation dataset (which contains about 20 percent of the whole dataset) and the test dataset (which contains 10 percent of the total dataset). We will use the training dataset to train the models, while the testing dataset will be used to assess the model's accuracy after training. Finally, we ran the dataset through the tokenization algorithm. Tokenization is the process of breaking down a phrase, sentence, or text document into smaller pieces, such as individual words.

4 Proposed Methodology

4.1 Bi-LSTM

The extraordinary capacity of LSTM to extract complex text information is crucial in text classification. Variable-length sequences benefit from the attention mechanism as well as a more even distribution of weight. We employed Word2vec weights that had been pre-trained with a large corpus, guaranteeing that the model was more accurate. The model's accuracy was evaluated using precision, recall, and the F1-score. For natural language processing (NLP) applications, the LSTM model plus a CNN proved to be unexpectedly successful. An increasing number of research have proven that sequence-based sentiment categorization may be achieved using LSTMs. The LSTM's accuracy is further hindered by its inability to distinguish between different relationships between document components. To solve this problem, we have used data preprocessing. Before being input into the model, the text was preprocessed. This includes, among other things, removing whitespace and unneeded words, converting other words to approximations, and minimizing duplicate words.

4.2 GRU

GRUs are a gating mechanism which is used in RNN. There is a problem with short-term memory in RNN. When a sequence become large enough, they face difficulty sending information to later time steps from earlier ones. RNNs may leave out essential information at the start when attempting to predict anything from a paragraph of text. During back propagation, the vanishing gradient problem impacts recurrent neural networks. To update a neural network's weights, gradients are used which are actually values. The vanishing gradient problem occurs when a gradient decrease as it propagates backwards in time. If a gradient value goes below a fixed level, for learning it will not be useful after that. In recurrent neural networks, layers that get a minor gradient update cease learning.

Those are often the first layers to show up. RNNs may forget what they have watched in longer sequences since these layers do not learn, resulting in a short-term memory. GRUs were created to address the issue of short-term memory.% accuracy.

4.3 CNN

A convolutional layer is a kind of feature map which is further connected with local regions. Each of them takes input according to the kernel and produces an output received from the kernel.

Pooling Layer: The pooling layer is mainly used to decrease the parameters and computational complexities of a network. It divides the data into subsections and returns the maximum value from a particular subsection.

Dense Layer: Dense Layer is nothing but a method to carry out linear operations in order to convert the input within some predefined weights. It takes an activation function as input.

We have used a CNN model with embedding layer in which the first 2nd layer will be convolutional layers and others are fully connected. The output from the last fully connected layer will be passed into a Sigmoid, ReLu, and Adam which will distribute data amongst three classes. Each layer will be based on 1D feature maps and it will try to extract specific features while holding spatial information.

4.4 Activation Function

An activation function is a crucial factor in machine learning, neural networks or in the field of artificial intelligence. The main job of an activation function is, it basically decides whether a neuron should be activated or not. Moreover, it defines the output of that node given an input or set of inputs.

ReLu: It is rectified linear activation function which is basically a piecewise linear function that would only produce output if the input is greater than zero or positive. Otherwise, it will produce the output as zero. Because of its easier training and performing better many neural networks use it as their default activation function.

Sigmoid: A sigmoid function is a special form of the logistic function. It has a domain from negative infinity to positive infinity and a range from 0 to 1. This function is monotonically increasing and continuous everywhere. It is an Activision function for neural network.

Adam: In training data, we have to update the network weights in a iterative way and for that we use Adam which is an optimization algorithm. It is different from the usual stochastic gradient descent procedure which was usually used before. It is used to speed up the gradient descent technique by taking into account the gradients' exponentially weighted average. Using averages accelerates the algorithm's convergence to the minima.

Dropout: To address overfitting problems, dropout was used after the first dense layer in this model at a ratio of 0.5.

GlobalMaxPooling: This is another form of pooling layer. At first, we take the size of the pool equal to the input size so that the output value is the maximum of the entire input.

5 Experiments and Results

As previously stated, we have applied three distinct models to our dataset. The performance of all models is rather satisfactory. Among these models, the accuracy of GRU and Bi-LSTM is almost same and superior to that of CNN. Nevertheless, we may differentiate between the accuracy of GRU and Bi-LSTM and choose the more accurate of the two. To be more precise, CNN, Bi-LSTM, and GRU achieved 84 percent, 89 percent, and 90 percent accuracy, respectively. As can be seen, the accuracy of GRU is comparable to that of Bi-LSTM, although it is still superior. Therefore, selecting the GRU model for the dataset is preferable. The classification report for CNN, Bi-LSTM, and GRU are shown consecutively so that they may be differentiated easily.

	precision	recall	f1-score	support
class 0	0.79	0.99	0.88	346
class 1	0.76	0.92	0.83	290
class 2	0.79	0.97	0.87	864
class 3	0.85	0.28	0.42	209
class 4	0.99	0.79	0.88	687
class 5	1.00	0.43	0.60	103
accuracy			0.84	2499
macro avg	0.86	0.73	0.75	2499
weighted avg	0.86	0.84	0.82	2499

Fig. 1. Classification Report of CNN

	precision	recall	f1-score	support
class 0	0.94	0.83	0.88	314
class 1	0.90	0.78	0.83	270
class 2	0.90	0.94	0.92	687
class 3	0.88	0.67	0.76	160
class 4	0.89	0.97	0.93	626
class 5	0.65	0.84	0.74	88
accuracy			0.89	2145
macro avg	0.86	0.84	0.84	2145
weighted avg	0.89	0.89	0.89	2145

Fig. 2. Classification report of Bi-lstm

	precision	recall	f1-score	support
class 0	0.90	0.87	0.88	314
class 1	0.87	0.87	0.87	270
class 2	0.95	0.90	0.92	687
class 3	0.77	0.94	0.85	160
class 4	0.92	0.96	0.94	626
class 5	0.85	0.73	0.79	88
accuracy			0.90	2145
macro avg	0.88	0.88	0.88	2145
weighted avg	0.91	0.90	0.90	2145

Fig. 3. Classification report of GRU

Additionally, we have created the confusion matrices for these models. Using a table called a confusion matrix, classification algorithms are assessed. It describes the classification model's performance. The confusion matrices of CNN, Bi-LSTM, and GRU are presented in the order shown below. In the figure, the rows and columns named from 0 to 5 correspond to the categories of emotions that we defined in our models.

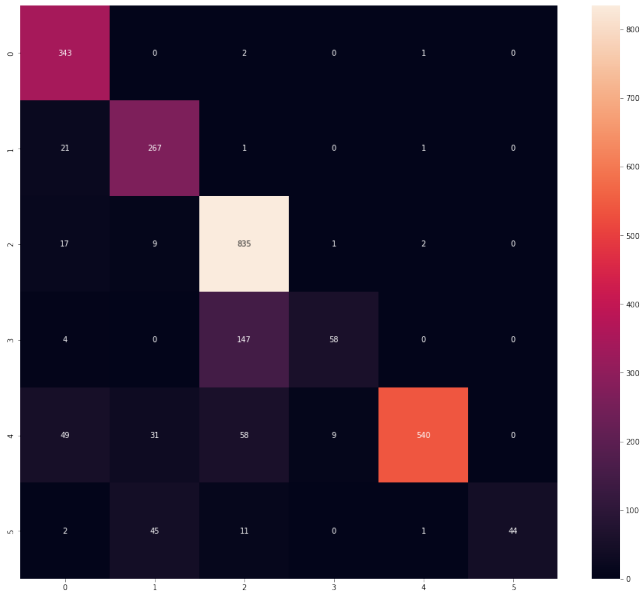

Fig. 4. Confusion Matrix of CNN

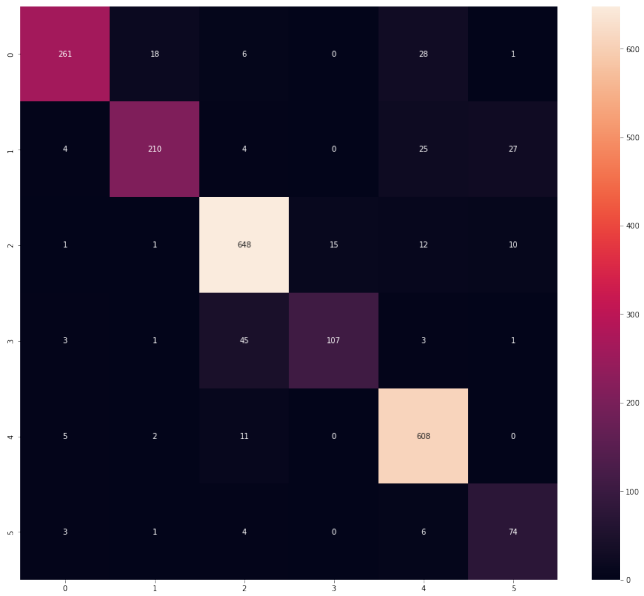

Fig. 5. Confusion Matrix of Bi-lstm

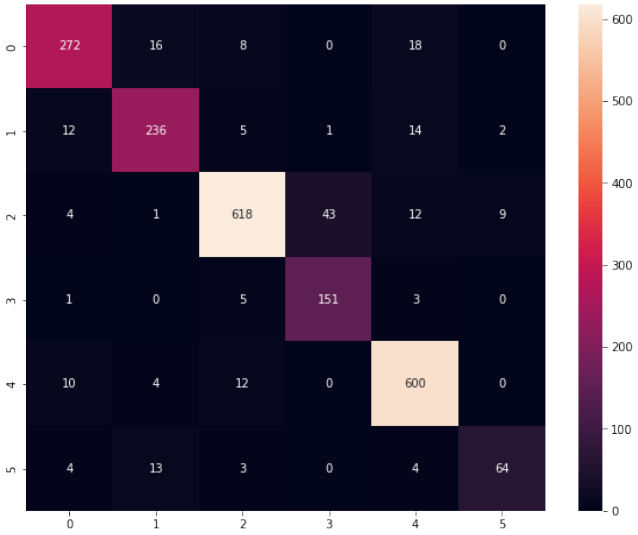

Fig. 6. Confusion Matrix of GRU

Because it is a relatively new concept, only a few frameworks exist to implement it. We utilized Flower for our paper. Flower is a new Federated Learning framework. Flower is designed to operate with all of them. It focuses on providing tools for effectively implementing Federated Learning while allowing you to focus on the training itself. It is really simple to set up a basic Federated configuration in Flower. In addition, the range of compatible devices is extremely broad, ranging from mobile devices to servers and others. As demonstrated in their paper, its design provides for scalability up to 1000s of clients[10]. We have used 100 clients and 10 federated rounds for our simulation. Though we received better accuracy in GRU compared to LSTM for centralized language modeling tasks from our dataset. It showed no difference in performance for federated learning simulation. We only got better accuracy after increasing the federated rounds. The GRU model has less gates compared to LSTM so it is a little speedier in training but it did not help for federated settings. The solution for a better accuracy in federated settings could be to increase the dataset and increasing the federated rounds without changing the model architectures.

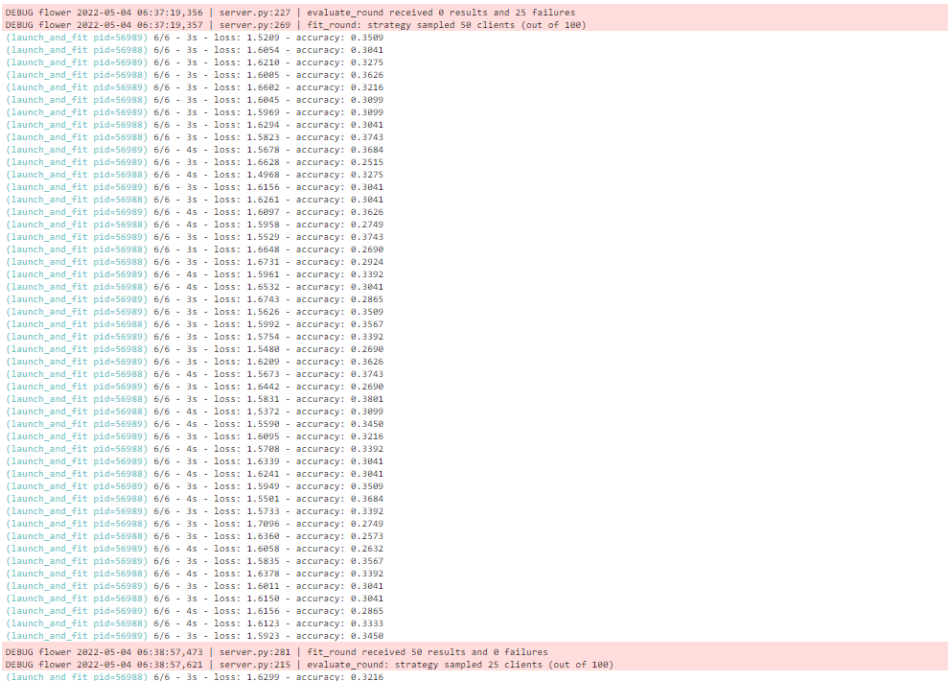

Fig. 7. Accuracy of individual clients for GRU

So in the end our three models accuracy are given bellow:

Model Name	Accuracy
CNN	84%
Bi-lstm	89%
GRU	90%

Table 1: Final accuracy of our three Models we have used in our experiment

6 Conclusion

This work discusses detecting emotion using deep learning models and a federated learning architecture. In addition, the top text classification models for usage in the Federated Learning architecture are displayed. A system like this may be used to assess people's emotions and propose movies and television shows. However, the performance of such a model is dependent on a number of aspects, including the quality of data and parameters. Dataset suppliers should also update their databases as soon as new data becomes available in order to augment models through communication loops. If all of these can be maintained, such an architecture can ensure a very high accuracy, similar to or even better than what we showed.

References

1. Few-shot learning text classification in Federated Environments. IEEE Xplore. (n.d.). Retrieved May 9, 2022, from https://ieeexplore.ieee.org/document/9588833
2. Liu, D. Miller, T. (2020, February 20). Federated pretraining and fine tuning of Bert using clinical notes from multiple silos. arXiv.org. Retrieved May 5, 2022, from https://arxiv.org/abs/2002.08562
3. Leroy, D., Coucke, A., Lavril, T., Gisselbrecht, T., amp; Dureau, J. (2019, February 18). Federated learning for keyword spotting. arXiv.org. Retrieved May 6, 2022, from https://arxiv.org/abs/1810.05512
4. Stremmel, J., amp; Singh, A. (1970, January 1). Pretraining Federated Text Models for next word prediction. SpringerLink. Retrieved May 6, 2022, from https://link.springer.com/chapter/10.1007/978-3-030-73103-8_34
5. Ge, S., Wu, F., Wu, C., Qi, T., Huang, Y., amp; Xie, X. (2020, March 25). Fedner: Privacy-preserving medical named entity recognition with Federated Learning. arXiv.org. Retrieved May 6, 2022, from https://arxiv.org/abs/2003.09288
6. Basaldella, M., Antolli, E., Serra, G., amp; Tasso, C. (1970, January 1). Bidirectional LSTM recurrent neural network for keyphrase extraction. SpringerLink. Retrieved May 6, 2022, from https://link.springer.com/chapter/10.1007/978-3-319-73165-0_18
7. Bhattacharyya, S. (2020, November 17). Author(Multi-class text) classification using bidirectional LSTM amp; Keras. Medium. Retrieved May 6, 2022, from https://medium.com/analytics-vidhya/author-multi-class-text-classification-using-bidirectional-lstm-keras-c9a533a1cc4a
8. Muthukumar, N. (2021). Few-shot learning text classification in Federated Environments. 2021 Smart Technologies, Communication and Robotics (STCR). https://doi.org/10.1109/stcr51658.2021.9588833
9. Stremmel, J., amp; Singh, A. (2020, August 17). Pretraining Federated Text Models for next word prediction. arXiv.org. Retrieved May 6, 2022, from https://arxiv.org/abs/2005.04828
10. Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K. H., Parcollet, T., de Gusmão, P. P. B., amp; Lane, N. D. (2022, March 5). Flower: A friendly federated learning research framework. arXiv.org. Retrieved May 6, 2022, from https://arxiv.org/abs/2007.14390