

Evaluating the machine learning models based on natural language processing tasks

Meeradevi¹, Sowmya B. J.², Swetha B. N.²

¹Department of Artificial Intelligence and Machine Learning, Ramaiah Institute of Technology, Bangalore, India

²Department of Artificial Intelligence and Data Science, Ramaiah Institute of Technology, Bangalore, India

Article Info

Article history:

Received Jun 6, 2023

Revised Oct 18, 2023

Accepted Dec 20, 2023

Keywords:

Gated recurrent unit

Language model

Long short-term memory

Natural language processing

Sentiment analysis

ABSTRACT

In the realm of natural language processing (NLP), a diverse array of language models has emerged, catering to a wide spectrum of tasks, ranging from speaker recognition and auto-correction to sentiment analysis and stock prediction. The significance of language models in enabling the execution of these NLP tasks cannot be overstated. This study proposes an approach to enhance accuracy by leveraging a hybrid language model, combining the strengths of long short-term memory (LSTM) and gated recurrent unit (GRU). LSTM excels in preserving long-term dependencies in data, while GRU's simpler gating mechanism expedites the training process. The research endeavors to evaluate four variations of this hybrid model: LSTM, GRU, bidirectional long short-term memory (Bi-LSTM), and a combination of LSTM with GRU. These models are subjected to rigorous testing on two distinct datasets: one focused on IBM stock price prediction, and the other on Jigsaw toxic comment classification (sentiment analysis). This work represents a significant stride towards democratizing NLP capabilities, ensuring that even in resource-constrained settings, NLP models can exhibit improved performance. The anticipated implications of these findings span a wide spectrum of real-world applications and hold the potential to stimulate further research in the field of NLP.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Meeradevi

Department of Artificial Intelligence and Machine Learning, Ramaiah Institute of Technology

Bangalore, India

Email: meera_ak@msrit.edu

1. INTRODUCTION

In the past, language models were primarily designed to execute tasks based on their training data, lacking the capacity for understanding akin to human cognition. Various experiments have been undertaken to explore the extent to which language models can approach human-like comprehension. The integration of neural networks, mirroring elements of the human brain, has notably contributed to enhancing language model performance. Historically, statistical language models were employed, primarily focused on predicting the subsequent words in a given sentence based on their training data. However, a significant leap in language model performance is achieved when these models are underpinned by neural networks. This integration of neural networks significantly broadens the spectrum of natural language processing (NLP) tasks that a language model can tackle [1]–[3]. A neural language model exhibits versatility in handling NLP tasks, spanning from straightforward to intricate challenges. Yet, even with these advancements, replicating the seamless proficiency of human cognition in machines remains an elusive goal. Achieving a level of performance where language models can emulate human capabilities fully remains a work in progress. There are steps in which we can perform NLP tasks using language models. The first step is to collect the data; the

data collected has to be free of not available data so the missing data has to be cleaned to see that the language models do not result in biased outputs.

After analysis, the data is collected in a way that the input variables given are appropriate and linked to that of the particular NLP task provided to the language model. After the data that is collected is analyzed then using the data collected the data is divided into training data and testing data. NLP is the ability of machines to understand the way a human can understand some language. There are many tasks where a machine can be used instead of human beings, as we all know it is very difficult to replace a human with a machine as a human brain can never be replaced with a machine. However, with help of different available technologies, there are so many tasks based on NLP, which are efficiently carried out by language models using different networks [4]–[6].

Numerous language models are present at the time; each language model has its pros and cons. Considering all the points and coming up with a language model that could perform the best for that particular task is a challenge. Different NLP tasks require focusing on different functions, and the language model that is suitable for the particular function is difficult to recognize without knowing the language model functionality. Training a language model on the datasets is challenging as it is time-consuming and finding out the model when the overfitting occurs is important to evaluate language models [7]. Long short-term memory (LSTM) is good at memory whereas gated recurrent network (GRU) is less complex than that LSTM as GRU has two gates that are reset gate and an update gate whereas LSTM has three gates that are forget gate, an input gate, and output gate. Therefore, the combination of less complex and long-term memory results in a model, which could result better than the LSTM and GRU individually. The GRU language model consists of only two gates, which results in a less complex language model when compared to that of LSTM, and as a result, takes lesser time to train and test the GRU language model when compared to that of the LSTM language model.

2. LITERATURE SURVEY

Many experiments have been carried out recently on language models, few experiments that highlight how the language models are trained, what language models learn, and how a language model performs. The internet movie database (IMDB) review dataset with 50 k movie reviews was used in the experiment. Hugging face transformers were used to train the GPT-2. The experiment was carried out on a PC. The reviews to be evaluated, the example reviews with sentiments and prompts were given as input to the GPT-2 language model. 'I really' was the prompt used in this article. GPT-2 generated positive or negative sentiment for the given review. 'I liked this movie' with positive sentiment and 'I really disliked this movie' with negative sentiment. The heat map was generated for a full-length review and the output text was given by GPT-2, by observing this heat map it was very clear that the text output given by GPT-2 had a very clear sentiment attached when compared to a full-length review. The output generated by GPT-2 was correct for 92% of reviews. This experiment could be performed on a large review dataset to increase the efficiency of the model [8]. The scaling properties of the language models were evaluated and examined so that these metrics could provide us with the language models, which are better in the particular field. The assumption of scaling properties tested on the language models could lead us to language models that have better performance. The language models with long-term memory were considered to have greater efficiency when compared to the statistical language models. Using neural networks in the language models can lead to better language models in every aspect as they result in long-term memory. LSTM are the ones, which were observed to be having long-term memory and leading to greater perplexity [9].

The methodology used here divides metrics into 5 major groups for evaluation i.e. syntactic style, part-of-speech usage, readability and complexity, prompt-based conditioning, and measure of coherence. Moreover, these metrics were analyzed using a least absolute shrinkage and selection operator (LASSO) based regression model and inter-metric correlation. The metric score of the content generated by the models was compared with the metric score of human written content; no model was close to human written content. However, the fine-tuned GPT-2 models and the transformer-extra long (XL) models were the ones that had lowest deviation in the metrics whereas the XL Net models had a high value of deviation [10]. Data imbalance is one of the major problems in classification, lack of a dataset belonging to any of the class labels could result in data imbalance and further lead to poor classification performance results. Here GPT-2 and LSTM language models were exploited, the first stage is to convert an imbalanced dataset to a balanced dataset. The generated text dataset was evaluated using metrics: word-overlap metrics and embedding-based metrics.

The experiment was performed on two different types of datasets sentence-level and document-level, for the sentence-level dataset testing the LSTM language model performed excellently whereas, for the document-level testing, LSTM performed poorly, the reason behind these results would be that LSTM networks can grasp the contextual dependencies only on small text and fail for longer texts. However, the

GPT-2 based text generation algorithm solved this bottleneck and gave excellent results, especially for document-level datasets [11]. Label bias, regency bias, and common token bias are three drawbacks of language models that lead them to be biased toward certain answers during few-shot learning. To overcome these pitfalls, a contextual calibration process was introduced. The contextual calibration process leads to improved accuracy in a given task; however, language models fail in-context learning [12]. Many researchers for a long time have observed that GPT does not perform well following natural language understanding tasks with fine-tuning. The experiment was conducted using two well-known natural language understanding benchmarks namely leave against medical advice (LAMA) knowledge probing and SuperGlue. On the SuperGlue benchmark, GPT-style models show equal performance as BERTs because of P-tuning [13].

Language models perpetuate the prospect of a sequence of words. Language models are trained in a way that they can differentiate between similar-sounding words and phrases. A statistical language model provides a probability distribution over the words being used to train the model. See that out of the entire well-known models available the one that would perform better is to be considered for the applications for humankind. Therefore, to come up with a model that could perform better we evaluate the models based on individual perplexity, the one, which stores the information for the long term, could be the one with the better outcomes. The question over here is how to evaluate these models: human rating, probabilistic distribution, and perplexity.

These were the methods used before to rate the models based on their results. Other metrics used for evaluating the model is bilingual evaluation understudy (BLEU) and recall-oriented understudy for gisting evaluation (ROUGE). The scaling properties of language models are used to come up with the conclusion of which language model is most appropriate for coming up with a correct result. The language models built on recurrent neural networks (RNN) with gating mechanisms have long memory behavior of the natural language. Scaling properties of language models can be evaluated using different laws for different properties, for example, Zipf's law inspects if the language model can fabricate uncommon words likewise Taylor's law inspects that a model can acquire long memory in natural language text [14]. The language models are evaluated using numerous techniques, the language models are being used in various applications. The goal is to collect data sources in Indic languages widely [14]. Different applications require different language models based on the expertise of the individual language model. To decide on which language model could the application be used in a better way the language model is to be evaluated and based on the evaluation result one could conclude the language model pertinent for a particular application. Most of the language models are used in NLP tasks, so another way of evaluating a language model could be testing the language models by assigning an individual model a particular task and comparing the result of individual language models. The language model with excellent results for that particular task could be considered in the application. Each language model has different features, which can be brought to bear in particular applications.

The evaluation can also be done from the datasets available on WordNet, ConceptNet [15], or BabelNet [16], [17]. The language models play a very vital role in NLP tasks; the language models can be evaluated based on the results generated by the language model for the particular task [18]. Different language models can be used for the same NLP tasks and the output of the language models can be compared to each other. The lesser the perplexity the better the performance of the language model. The language model that has a tighter fit to the test data is the one, which has better probability and is the better model. One of the NLP tasks on which the language model can be evaluated is autocorrect software, which automatically suggests the correct word that could be mistyped by a human while writing any document. It works as: i) identify the misspelled word, ii) find strings n edit distance away, iii) filter candidates, and iv) calculate word probabilities.

The algorithm used to see that the autocorrect system works smoothly and efficiently is the minimum edit distance algorithm [19]. The two approaches to evaluate the language models. Evaluation of the natural language's models such as statistical language models and the natural language models is performed in the paper: i) straightly examine the part of the language model and ii) substantiate the result generated by the model.

To break through NLP tasks, neural language models play's indispensable role. The throughput of each model is assessed on perplexity, now we assess N-gram language models, neural language models, probabilistic context free grammar (PCFG) language models based on Simon/Pitman-Yor processes. Generative adversarial networks (GAN) for text generation under the scrutiny of five scaling properties given by Taylor's law, Zipf's law, Heaps' law, long-range correlation analysis, and Ebeling's method. There are numerous NLP tasks where language models are used, to see that the application is carried out efficiently it is very important to choose a language model that works up to the mark and produces the results accordingly. Therefore, evaluating these language models plays a vital role in further applications at a high level. These were the metrics used earlier for evaluating the language models. The examination of language models built on probability distribution is said to be a perplexity-based evaluation. This weighs the exactness of the probability of the words. $\text{Perplexity} = f(r)\alpha r^{-\alpha}$ [14]. Perplexity is considered the standard automated metric

for examining the model quality, so the other metrics used to evaluate the models are weighed up with perplexity. The metrics BLEU/ROUGE are orthodoxly used in paired-corpus-oriented tasks like machine translation [15]. This evaluation is done only based on the output of the language model and does not require any access to the internal elements. The evaluation of the models using these methods remains questionable as it was introduced to differentiate between the results produced by different language models.

Evaluating language models independent of the model distribution or a reference can be done by using different language models as a referral to evaluate the current model result. PCFG is another method of evaluating language models without any reference. Here scaling properties of the language models were evaluated and examined so that these metrics could provide us with the language models, which are better in the particular field [20]. The assumption of scaling properties tested on the language models could lead us to the language models that have better performance. The language models with long-term memory were considered to have greater efficiency when compared to the statistical language models. Using neural networks in the language models can lead to better language models in every aspect as they result in long-term memory. LSTM are the ones, which were observed to be having long-term memory and leading to greater perplexity [16].

There are many language models, which are available today. Each language model has its pros and cons. This language model can be evaluated to check its efficiency and use the language model accordingly. The language models, which we test in this article, are N-gram language models, grammatical language models, and neural language models, which can be evaluated on the standard benchmarks [21]. N-gram (N is a sequence of words) language models are the elemental models. There is unigram, bigram, and many, which vary with the number of words, dealt with. In this article, we deal with “3-gram and 5-gram language models” [22]. The N-gram model could be evaluated with various NLP tasks like fill in the blanks, and autocomplete where the N-gram model predicts the next word based on the type of gram, which is being used. The model is trained using the data. The basic grammatical model is PCFG. This model produces text using three neural language models chiefly acquiring RNN [23]. RNN helps models create a long memory, which helps, in improved perplexity. RNNs with gating mechanisms such as “LSTM, quasi-recurrent neural network (QRNN), and GRU” are generally acquired [24]. GAN abridge generative adversarial network, these models produce new data, which is relevant to the dataset used to train the language model. The data that is produced by the GAN models can be compared with the human written data and the efficiency of the produced data could be tested. The more the data produced by the GAN model matches the human written data the better the model and the results in less perplexity [25], [26].

3. DESIGN AND IMPLEMENTATION

The language models are trained by feeding in the training data. The language models are given in the testing data and the language models produce the output based on the training given, this output is evaluated based on the standard metrics of language models. Similar to the NLP task output can also be compared and evaluated.

The first step as shown in Figure 1, the first step is to collect the data, after collecting the data, the data is split into two that is training data and testing data. Where training data is used to train the language model and the learning rate is given with which a model is trained from that learning the model is next tested using test data that is how it is important to have test and train data to evaluate a language model. Initially, the available language model that is LSTM, bidirectional long short-term memory (Bi-LSTM), and GRU were trained and tested comparing the results of these three language models the conclusion was made that the LSTM and the GRU were top on the list. That is why a hybrid model that is LSTM with GRU was implemented to build a language model, which could result in even better performance. The performance of a language model was measured in the form of a regression graph and the error was calculated using root mean square error. The language model, which produces the least error, is the one, which is said to be the best language model. When language models are evaluated on the same task using the same dataset it makes it easier to come up with the best model among the tested.

Collect the data and then the dataset collected is split into two that is one as a training dataset and the other as a testing dataset. The dataset cleaning is also important as any missing data in the dataset could affect the results of the testing. Once the dataset is ready, the predicting model is designed using the neural network and the dropout method. Once the predicting model is built, the dataset is used to train the predicting model. The research shows using the neural network to build a predicting model is a better option as the neural networks are better at performance in predicting tasks. Based on the trained data the predicting language model predicts the stock price for the test data given as input. The output of the predicted model is in the regression analysis form using which the results of different predicting models can be compared and the conclusion of the best models can be made.

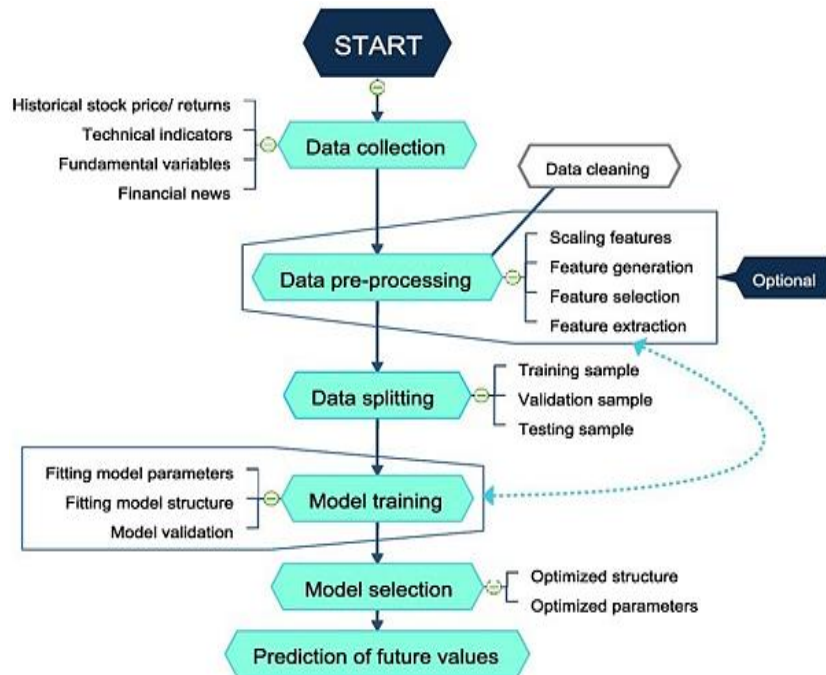


Figure 1. Evaluation of language model

The square root mean error or the accuracy of each predicting language model can be taken to get the difference between the predicted stock price and the real stock price. The lesser the error is the better the predicting language model is at the predicting task. All the predicting language models used are compared based on the accuracy, the root mean square error, and the predicting language models are compared on their performance, and the conclusion of the best model is done. This way the language model can be evaluated and can be compared on the same NLP task on the same dataset using the same methods as their output form. The jigsaw toxic comment classification is the other dataset on which the language models are trained and tested, where the output is given in the form of accuracy and is used to compare the different trained and tested language models. Sentimental analysis is the NLP task that is to be done using the language models.

3.1. The proposed model of long short-term memory integrated with the gated recurrent unit

The LSTM layers and the GRU layers are put together and a hybrid model is created as the LSTM language model and the GRU language model were the language models, which gave the least error when the evaluation was carried out. The root means square error that LSTM gave was 2.6790977261067916 whereas the root means square error that GRU gave was 2.0884588214463875. The root means square error that Bi-LSTM gave was 2.834516833852457 as the LSTM and GRU were the ones to give less error in the hybrid model of LSTM. The GRU layer could produce a much lesser error, so the layers of LSTM and GRU were taken together to create a better model which could give lesser error and predict the stock price more accurately. Whereas for the dataset Jigsaw toxic comment classification the LSTM with GRU model gave better accuracy when compared to the other language models evaluated. Figure 2 represents the loaded LSTM with GRU model for the IBM stock price prediction whereas Figure 3 represents the loaded LSTM with GRU model for the jigsaw toxic comment classification.

Figure 2 shows the loaded LSTM with GRU language model for the IBM stock price prediction has four dropouts and one dense layer as this IBM stock price prediction dataset has a smaller data size. Whereas the loaded LSTM with GRU language model for the Jigsaw toxic comment classification dataset has two dense layers and has no dropouts as the data size of the Jigsaw toxic comment classification is large. The language model when being trained on the large dataset needs denser layers whereas, for the smaller dataset, the layers do not need to be that dense. The language models take more time when being trained using denser layers and if the data size is large. The dense layer is nothing but the neurons that are interconnected with each other to compute the weighted average of the input. The dense layer is used in applications such as classification and prediction tasks that are NLP tasks.

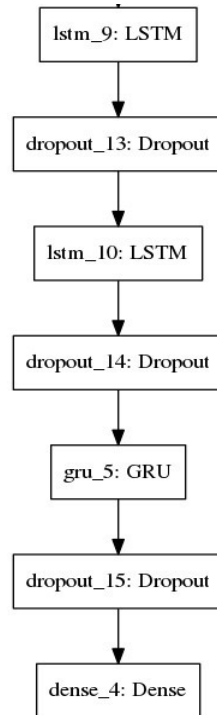


Figure 2. Loaded LSTM with GRU model for stock price prediction

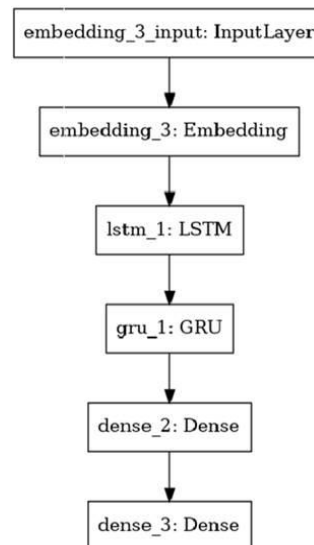


Figure 3. Loaded LSTM model for Jigsaw toxic comment classification

4. RESULTS AND INFERENCES

4.1. Exploratory data analysis for ibm stock prediction dataset

There are two different datasets on which these language models are evaluated, one is based on stock price prediction and another is based on the Jigsaw comments dataset. The stock price data is collected on daily bases so a data size of 12 years is used to train and test the model, which is split into two for testing and training. Ten years of data are used to train the model and two years of data to test the models. The dataset sample is shown in Figure 4.

There are many factors based on which the future stock price can be predicted like volume, price, and return keeping all of the factors into count the further stock price can be predicted. The Figure 4 shows the open, high, low, close, volume, and date which are the factors of the IBM stock price given as the input to

the language models, based on which the language models can predict the further stock price. The data is collected on a daily bases using which further prices can be predicted.

Figure 5 shows the split for training the language model and testing the language model for the IBM stock price data. The testing data that is used is two years of data and for training a language model eight years of data is considered. The LSTM language model, GRU language model, Bi-LSTM language model, and LSTM with GRU language model are the language models, which have been trained in this experiment. These language models are trained on IBM stock prices for ten years that is from 2006 to 2016, are tested on the stock prices, and are tested in 2017 and 2018. The stock price dataset is a time series dataset as the day-to-day entry is taken into account to picture the difficulties.

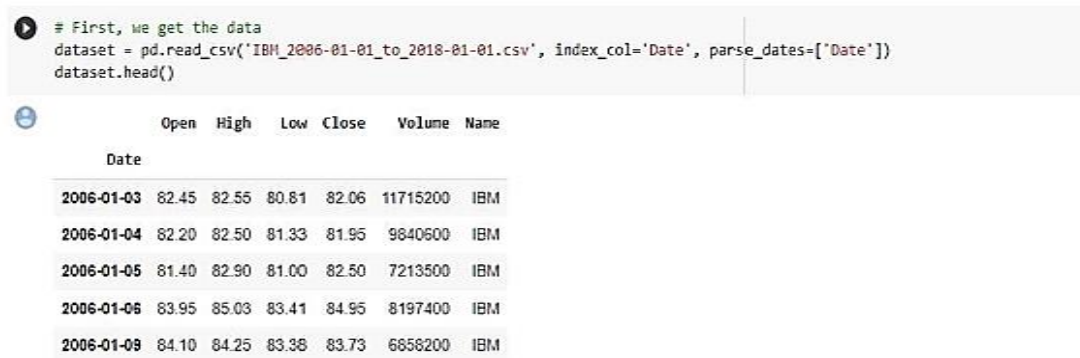


Figure 4. Sample of data set



Figure 5. Representation of data used

As seen in the figure the stock price has been on its top in the year 2013 and has been lowest in 2006. The price of stocks depends on multiple factors, so having to train and test a language model, it is very important to consider all the important factors into count to have a language model that can predict the future stock price most accurately. If the training consists of many missing values, then the language model could not be trained at its best and could result in biased outputs or accuracy. The training data plays a very important role to evaluate any of the language models.

Figure 5 represent the IBM stock price data that is used to train and test the language models. Figure 6 represents the price of the stock of the years 2010, 2011, 2012, and 2013 in January, May, and September. There are difficulties in the price from year to year. Figure 7 represents the return on stocks. The return on the stock is used to know the performance of the stocks and is an important factor to be considered to predict the stock price with the help of language models. Figure 8 represents the volume at which the stock price has been sold and bought; it plays a very important role to predict the stock price for later days. Figure 9 represents the IBM stock closing prices, which are given as input to language models to predict the stock prices. The opening prices and closing prices both are important factors so both of these are provided as the inputs to the language models to be trained on it and predict the further prices. The language models are trained using the learning rate, which decides the accuracy of the tested language model.

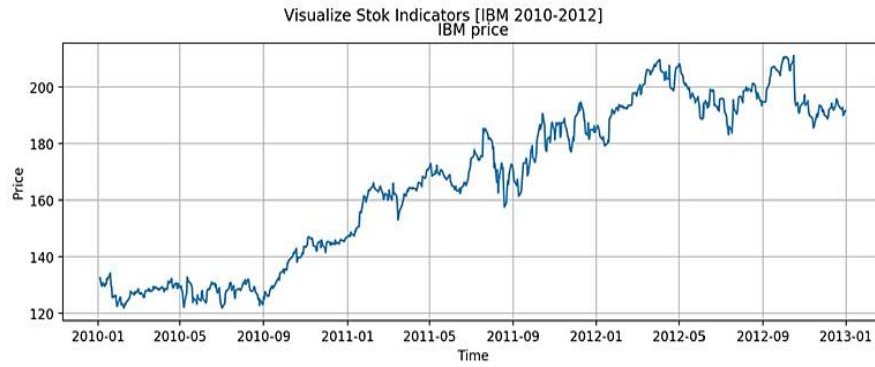


Figure 6. Visualize stock indicators [IBM 2010-2012] IBM price

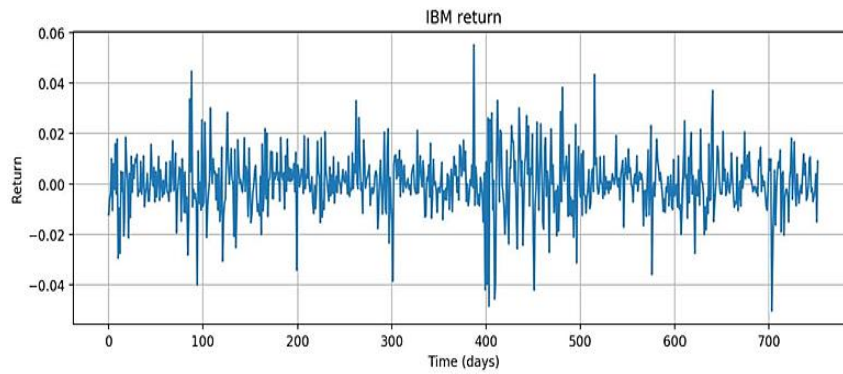


Figure 7. Visualize stock indicators [IBM 2010-2012] AIBM return

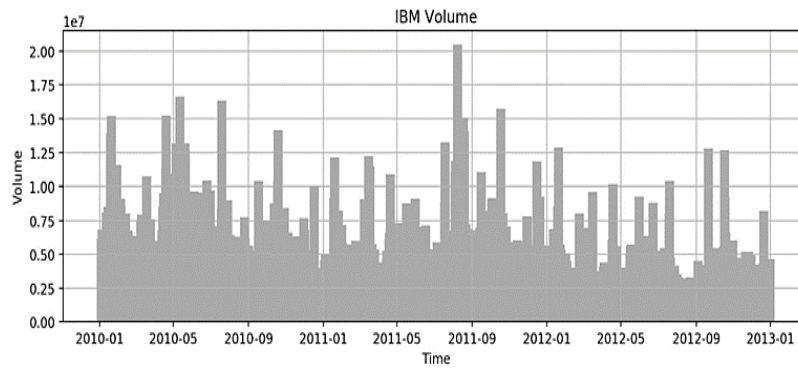


Figure 8. Visualize stock indicators [IBM 2010-2012] IBM volume

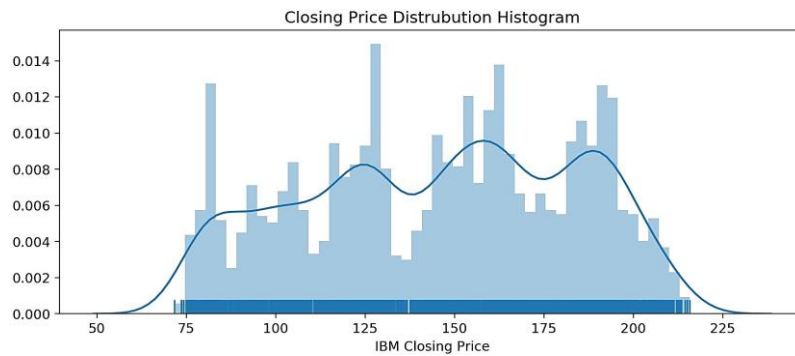


Figure 9. IBM stock closing prices

4.2. Exploratory data analysis for jigsaw toxic comment dataset

The other dataset that is used to evaluate language models here is the Jigsaw comment dataset. The size of the dataset is 153164 rows×6 columns, which is a large dataset. The dataset has multilingual comments. The Jigsaw toxic comment dataset sample is shown in Table 1, the comments are classified into six categories identity hate, insult, threat, obscene, severe toxic, and toxic, the zero represents that the comment is non-toxic. Figure 10 represents a pie chart for the distribution of the dataset over comments. Figure 11 shows the pie chart for the non-English language that is present in the dataset.

Table 1. Jigsaw toxic comment dataset sample

id	Comment_text	Toxic	Severe_toxic	Threat	Insult	Identity_hate
0000097843	Explanation edits make username hardcore metal...	0	0	0	0	0
0000814356	Aww match background color seemingly stick.....	0	0	0	0	0
0000823569	Hey man really try edit war guy constantly...	0	0	0	0	0
0000356891	Make real suggestion improvement wonder sectio...	0	0	0	0	0
0031590011	Sir hero chance remember page	0	0	0	0	0

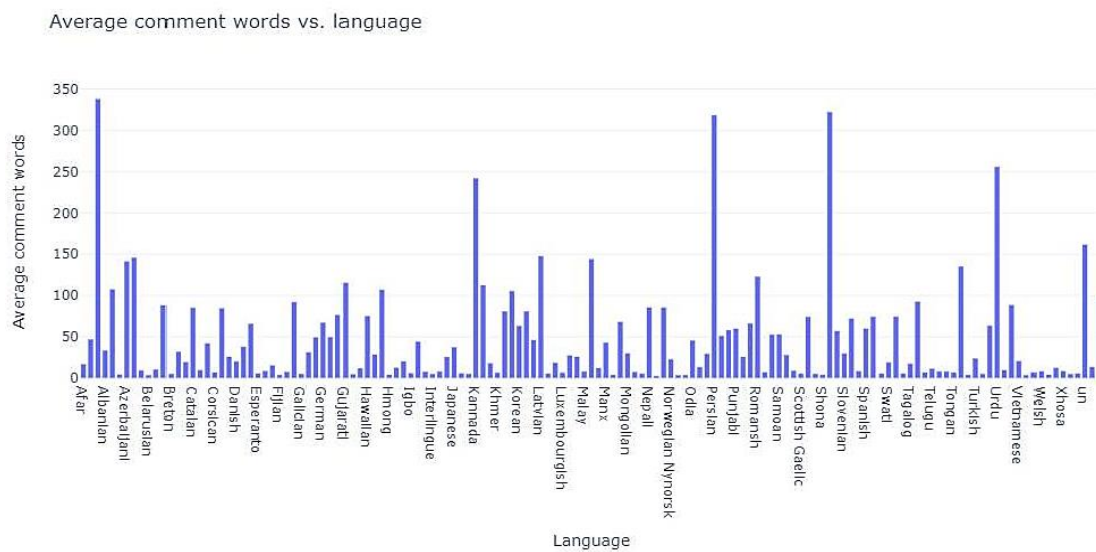


Figure 10. Average comment words vs Language

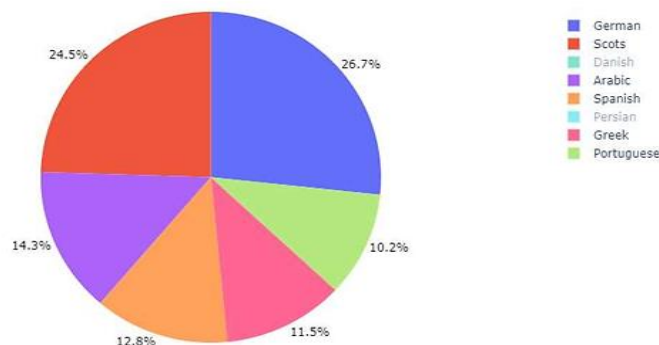


Figure 11. Pie chart of non-English language

Figure 12 shows the distribution of dataset over comments. Figure 13 shows the number of words in each comment. The jigsaw toxic comment dataset consists of toxic comments, which consist of six categories those are toxic, severe toxic, obscene, threat, insult, and identity hate. Figure 14 represents the correlation

matrix for different categories that are used for training and testing the language models. The correlation matrix is a way to represent the large dataset and how they are linked with each other. Table 1 represents how these 6 different categories that are identity hate, insult, threat, obscene, severe toxic, and toxic are linked to each other. The positive value indicates that the variables are linked with each other and if the value is negative then the variables are not linked to each other. Table 1 shows correlation matrix shows that all six categories have a link with each other and the language models are trained in a way that the language models can differentiate between these categories and identify the categories when they are tested by using new comments. Figures 10 to 14 are the graphs, which represent the performance of each language model.

Label distribution over comments (without "none" category)

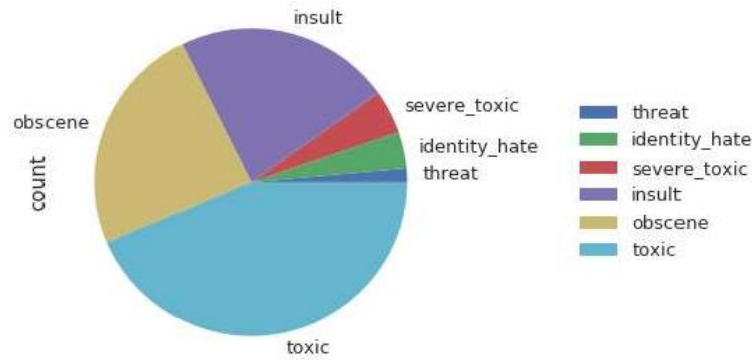


Figure 12. Distribution of dataset over comments

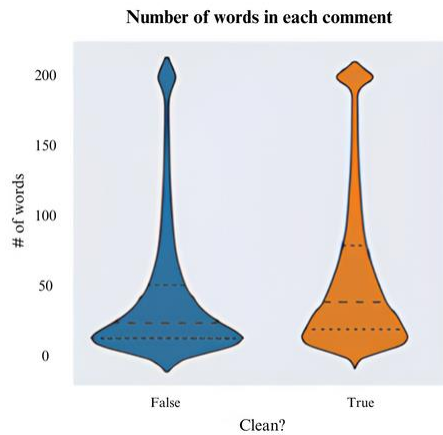


Figure 13. Number of words in comment

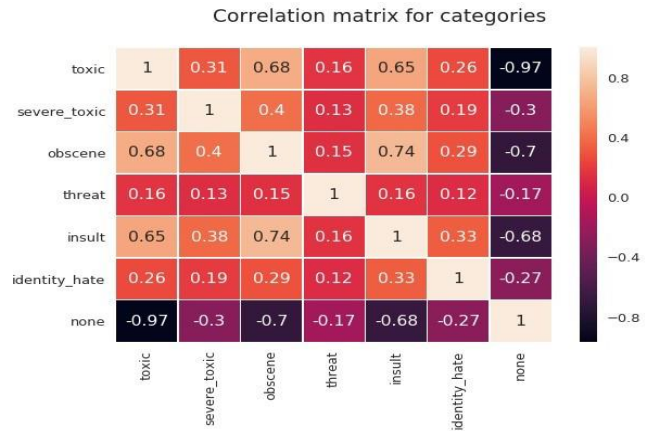


Figure 14. Correlation matrix for categories

4.3. Long short-term memory model

Figure 15 shows the LSTM language model performance for stock price prediction. The blue regression line indicates the predicted stock price whereas the red regression line indicates the real stock price. The error between these is calculated using the mean root squared error. The root means square error given by LSTM is 2.67.

4.4. Gated recurrent unit model

Figure 16 shows the performance of the GRU language model. The blue regression line indicates the predicted stock price whereas the red regression line indicates the real stock price. The error between these is calculated using the mean root squared error. The root means square error given by GRU is 2.08.

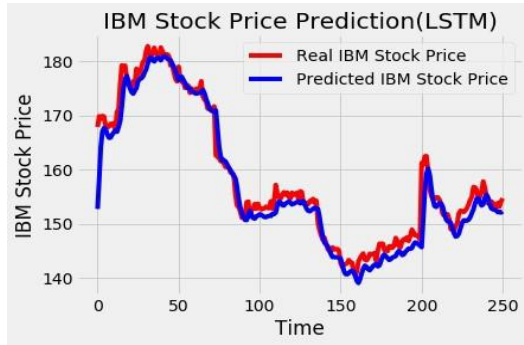


Figure 15. LSTM stock price prediction

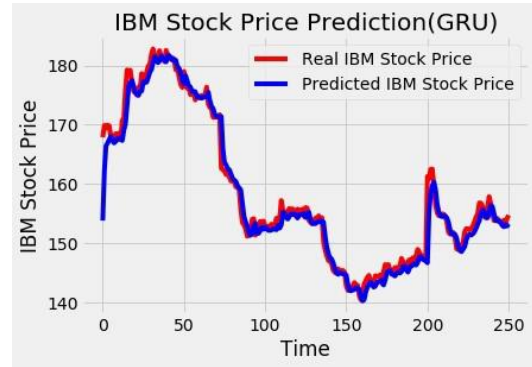


Figure 16. GRU stock price prediction

4.5. Bidirectional long-short-term memory model

Figure 17 shows the performance of the Bi-LSTM language model. The blue regression line indicates the predicted stock price whereas the red regression line indicates the real stock price. The error between these is calculated using the mean root squared error. The Bi-LSTM model did not perform as well as that of LSTM and GRU as seen in the Figure 17 the difference between that of predicted stock price and real stock price is more, so is the error which means the performance is that good in comparison to other models for this particular NLP task. The root means square error given by Bi-LSTM is 2.83.

Figure 18 shows the performance of the LSTM with the GRU language model. The blue regression line indicates the predicted stock price whereas the red regression line indicates the real stock price. The error between these is calculated using the mean root squared error. The Figure 18 represents that the difference between the predicted stock price and that of the real stock price is lesser, which means that the root square error is lesser and the performance is better. When compared to all the other evaluated language models the LSTM with GRU model performed better, resulting as the best of all the evaluated language models. The root means square error given by LSTM with the GRU model is 1.91.

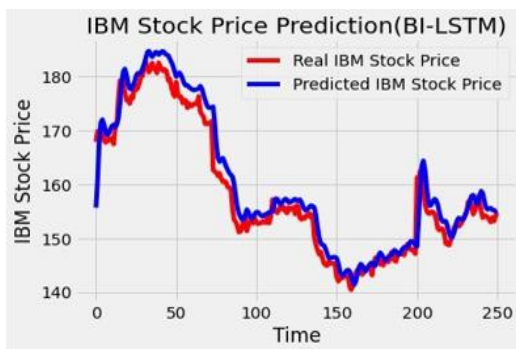


Figure 17. Bi-LSTM stock price prediction

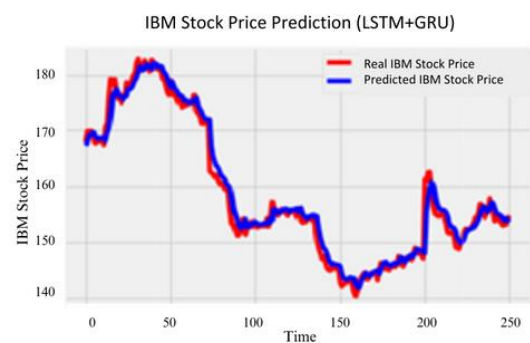


Figure 18. LSTM with GRU stock price prediction

4.6. Jigsaw toxic comment dataset

The language models LSTM, Bi-LSTM, GRU, and LSTM with GRU all these models were evaluated based on the same dataset for the same NLP task, which is sentiment analysis. Figures 19 to 26 are the graphs, which represent the performance of each language model. The outputs are shown for the Jigsaw toxic comment dataset. The outputs are given in the form of training and validation loss and training and validation accuracy. The output graphs of each language model are compared with each other to come up with the model, which results in the best performance. The jigsaw toxic comment classification dataset is split into a training dataset and a testing dataset, where 20% dataset is taken as the testing dataset and 80% of the dataset is taken as the testing data. The data size of the jigsaw toxic comment dataset is 153164 rows×6 columns, which is considered to be a large dataset. The outputs of the language models may differ from Figures 17 and 18, as the IBM stock price prediction dataset is smaller whereas the jigsaw toxic comment

classification dataset is large. This is due to the reason that the different language models are designed in some specific way, which was according to their needs now.

The difference between the accuracy of the language models that is LSTM, GRU, Bi-LSTM, and LSTM with GRU is not that large. However, the accuracy given by the language model that is LSTM with GRU is better when compared to others, which means that the hybrid model of LSTM and GRU performed better as the less complex gates of GRU. LSTM together increased the accuracy of the LSTM with GRU hybrid language model.

4.7. Proposed model long short-term memory integrated with gated recurrent unit model

The LSTM with GRU language model was trained and tested on the jigsaw toxic comment classification for 50 epochs. Figure 19 represents the training and validation loss for the LSTM with GRU model. Figure 20 represents the training and validation accuracy for the LSTM with GRU model.

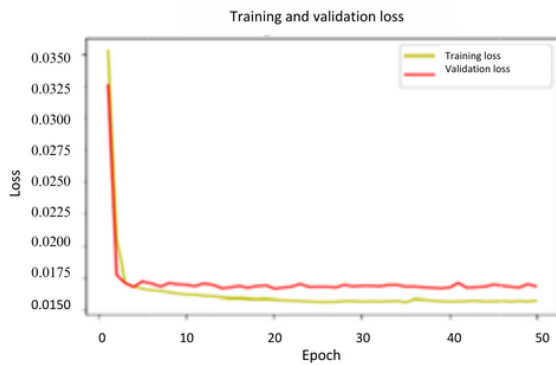


Figure 19. Training and validation loss for LSTM with GRU

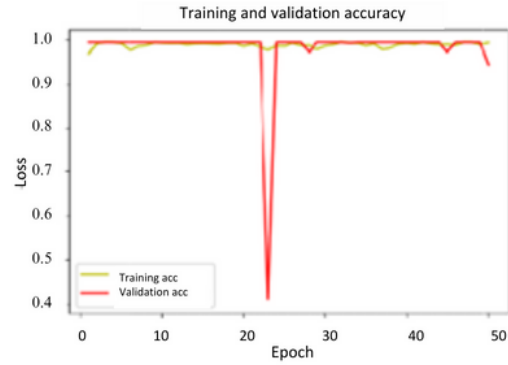


Figure 20. Training and validation accuracy for LSTM with GR

4.8. Bidirectional long-short-term memory model

The Bi-LSTM language model was trained and tested on the jigsaw toxic comment classification for 50 epochs. Figure 21 represents the training and validation loss for the Bi-LSTM model. Whereas Figure 22 represents the training and validation accuracy for the Bi-LSTM model. Figure 23 shows the training and validation loss for LSTM. Figure 24 shows the training and validation accuracy for LSTM.

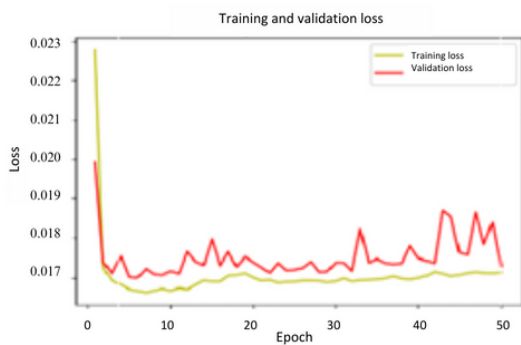


Figure 21. Training and validation loss for Bi-LSTM

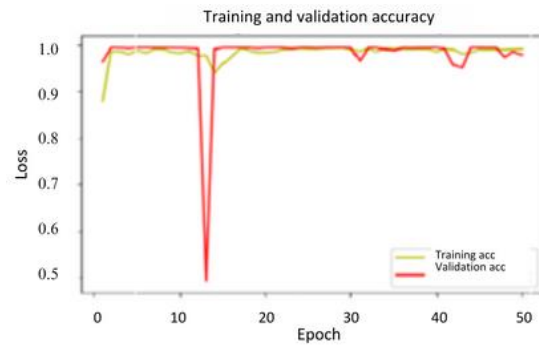


Figure 22. Training and validation accuracy for Bi-LSTM

4.9. Long short-term memory model

Since a typical RNN just has one hidden state that is transferred across time, learning long-term dependencies may be challenging for the network. In order to solve this issue, LSTMs introduce memory cells, which are long-term information storage units.

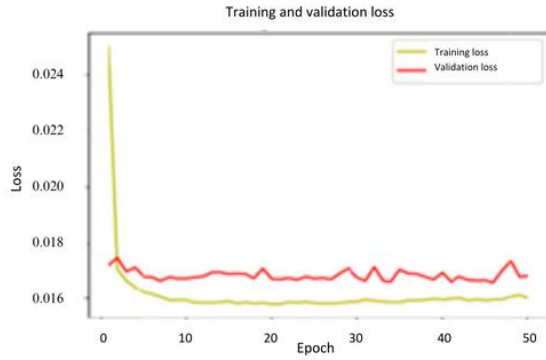


Figure 23. Training and validation loss for LSTM

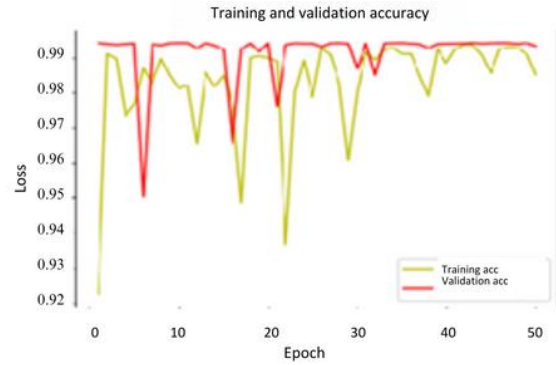


Figure 24. Training and validation accuracy for LSTM

4.10. Gated recurrent unit model

The GRU language model was trained and tested on the jigsaw toxic comment classification for 50 epochs. Figure 25 shows the training and validation loss for the GRU model. Figure 26 represents the training and validation accuracy for the GRU model.

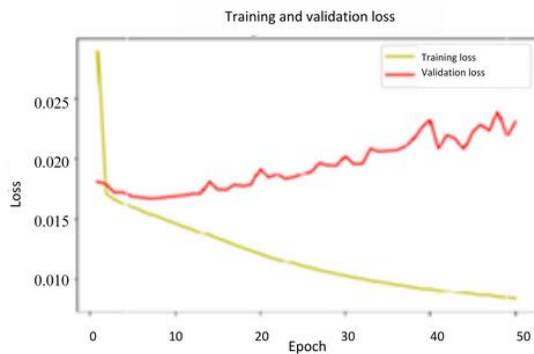


Figure 25. Training and validation accuracy for GRU

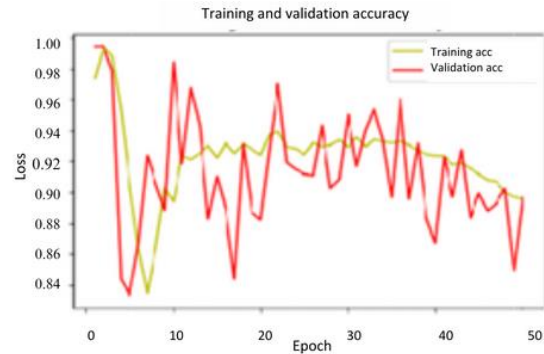


Figure 26. Training and validation loss for GRU

4.11. Table of comparison

4.11.1. Table for stock price prediction dataset

Table 2 is a comparison table, which represents the performance of all the evaluated models. The lesser the root mean square error value the better the model. As seen Table 2 makes it very clear that the LSTM with GRU is the language model which performed the best followed by the GRU language model followed by the LSTM language model and the Bi-LSTM language model gave more error compared to others.

Table 2. Table of comparison for stock price prediction dataset

Language models	Root mean square error
LSTM integrated with GRU	1.91
Bi-LSTM	2.67
LSTM	2.08
GRU	2.83

4.11.2. Table for jigsaw toxic comment classification

Table 3 is a comparison table, which represents the performance of all the evaluated models. The better, the accuracy the better the performance. As seen Table 3 makes it very clear that the LSTM with GRU is the language model which performed the best followed by the Bi-LSTM language model followed by the LSTM language model and the GRU language model gave lesser accuracy compared to others.

Table 3. Table of comparison for Jigsaw toxic comment classification dataset

Language models	Accuracy (%)
LSTM integrated with GRU	97.61
Bi-LSTM	96.67
LSTM	95.98
GRU	94.97

5. CONCLUSION

The LSTM, GRU, Bi-LSTM, and LSTM with GRU were the models that were evaluated and the LSTM with GRU model resulted as the best model which gave the least mean square root error that is 1.9110572761711948 for IBM stock price prediction data and LSTM with GRU model gave a better result for Jigsaw toxic comment classification dataset. More language models can be used to evaluate and more language models can be compared to come up with the best language model. For the stock price, prediction, which is a small dataset size the GRU language model, gave better results and for the sentiment analysis task, the language model GRU did not perform the best. Therefore, it is observed that the GRU language model performs better with small datasets when compared to the large dataset and the other observation that is made is that the Bi-LSTM language model performed better with the large dataset when compared to the small dataset. The future work that can be carried out is language models that perform better for a specific dataset can be combined and a hybrid model can be created to check if it gives out better results.





REFERENCES

- [1] D. Song, S. Gao, B. He, and F. Schilder, "On the effectiveness of pre-trained language models for legal natural language processing: an empirical study," *IEEE Access*, vol. 10, pp. 75835–75858, 2022, doi: 10.1109/ACCESS.2022.3190408.
- [2] O. Sen *et al.*, "Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods," *IEEE Access*, vol. 10, pp. 38999–39044, 2022, doi: 10.1109/ACCESS.2022.3165563.
- [3] P. Danenas and T. Skersys, "Exploring natural language processing in model-to-model transformations," *IEEE Access*, vol. 10, pp. 116942–116958, 2022, doi: 10.1109/ACCESS.2022.3219455.
- [4] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021, doi: 10.1109/TNNLS.2020.2979670.
- [5] Q. A. Shredha and A. A. Hanani, "Identifying non-functional requirements from unconstrained documents using natural language processing and machine learning approaches," *IEEE Access*, vol. 4, pp. 1–22, 2021, doi: 10.1109/ACCESS.2021.3052921.
- [6] Y. H. Chuang *et al.*, "Effective natural language processing and interpretable machine learning for structuring ct liver-tumor reports," *IEEE Access*, vol. 10, pp. 116273–116286, 2022, doi: 10.1109/ACCESS.2022.3218646.
- [7] M. Biswas, A. Shome, M. A. Islam, A. J. Nova, and S. Ahmed, "Predicting stock market price: A logical strategy using deep learning," *ISCAIE 2021 - IEEE 11th Symposium on Computer Applications and Industrial Electronics*, pp. 218–223, 2021, doi: 10.1109/ISCAIE51753.2021.9431817.
- [8] A. Panchbhai and S. Pankanti, "Exploring large language models in a limited resource scenario," *Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering*, pp. 147–152, 2021, doi: 10.1109/Confluence51648.2021.9377081.
- [9] S. Takahashi and K. Tanaka-Ishii, "Evaluating computational language models with scaling properties of natural language," *Computational Linguistics*, vol. 45, no. 3, pp. 481–513, Sep. 2019, doi: 10.1162/coli_a_00355.
- [10] A. Das and R. M. Verma, "Can machines tell stories? A comparative study of deep neural language models and metrics," *IEEE Access*, vol. 8, pp. 181258–181292, 2020, doi: 10.1109/ACCESS.2020.3023421.
- [11] S. Shaikh, S. M. Daudpota, A. S. Imran, and Z. Kastrati, "Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models," *Applied Sciences*, vol. 11, no. 2, pp. 1–20, 2021, doi: 10.3390/app11020869.
- [12] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," *Proceedings of Machine Learning Research*, vol. 139, pp. 12697–12706, 2021.
- [13] X. Zhou, Y. Zhang, L. Cui, and D. Huang, "Evaluating commonsense in pre-trained language models," *34th AAAI Conference on Artificial Intelligence*, pp. 9733–9740, 2020, doi: 10.1609/aaai.v34i05.6523.
- [14] X. Liu *et al.*, "GPT understands, too," *AI Open*, pp. 1–11, 2023, doi: 10.1016/j.aiopen.2023.08.012.
- [15] D. Kakwani *et al.*, "IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, pp. 4948–4961, 2020, doi: 10.18653/v1/2020.findings-emnlp.445.
- [16] O. Sainz and G. Rigau, "Ask2Transformers: Zero-shot domain labelling with pre-trained language models," *GWC 2021 - Proceedings of the 11th Global Wordnet Conference*, pp. 44–52, 2021.
- [17] N. Kassner, P. Dufter, and H. Schütze, "Multilingual LAMA: Investigating knowledge in multilingual pretrained language models," *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 3250–3258, 2021, doi: 10.18653/v1/2021.eacl-main.284.
- [18] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, pp. 4444–4451, 2017, doi: 10.1609/aaai.v31i1.11164.
- [19] E. H. Houssein, R. E. Mohamed, and A. A. Ali, "Machine learning techniques for biomedical natural language processing: a comprehensive review," *IEEE Access*, vol. 9, pp. 140628–140653, 2021, doi: 10.1109/ACCESS.2021.3119621.
- [20] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012, doi: 10.1016/j.artint.2012.07.001.
- [21] S. R. Bowman and G. E. Dahl, "What will it take to fix benchmarking in natural language understanding?," *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pp. 4843–4855, 2021, doi: 10.18653/v1/2021.naacl-main.385.
- [22] K. Lakhotia *et al.*, "Generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, Feb. 2021, doi: 10.1162/tacl_a_00430.





- [23] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1-10, 2018.
- [24] R. Marvin and T. Linzen, "Targeted syntactic evaluation of language models," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pp. 1192–1202, 2018, doi: 10.18653/v1/d18-1151.
- [25] P. S. Huang *et al.*, "Reducing sentiment bias in language models via counterfactual evaluation," *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, pp. 65–83, 2020, doi: 10.18653/v1/2020.findings-emnlp.7.
- [26] H. Jelodar, Y. Wang, R. Orji, and S. Huang, "Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2733–2742, 2020, doi: 10.1109/JBHI.2020.3001216.

BIOGRAPHIES OF AUTHORS







Meeradevi     working as an associate professor in Department of Artificial Intelligence & Machine Learning of Ramaiah Institute of Technology. She has 18 years of working experience. She has done her B.E. in 2006, and M.Tech. in 2009 from VTU. Her area of interests is wireless sensor network, computer security, machine learning, and computer networks. She can be contacted at email: meera_ak@msrit.edu.



Sowmya B. J.     working as an associate professor in Department of Artificial Intelligence and Data Science of Ramaiah Institute of Technology. She has 14 years of experience. She has done her M.Tech. in 2013 and Ph.D. in 2022. Her area of interests is deep learning, data analytics, software engineering, and machine learning. She can be contacted at email: sowmyabj@msrit.edu.



Swetha B. N.     is working as an assistant professor in Department of Artificial Intelligence and Data Science of Ramaiah Institute of Technology. Her areas of interest include machine learning, deep learning, and data analytics. She can be contacted at email: swethabn@msrit.edu.