**Twitter Tweet Spam Classification**

**Harshini Badam, Sathwik Vattikuti, Mandhakini Nallapaneni**

**Abstract**

With the exponential growth in Online Social networking particularly Twitter, Facebook and Instagram, these platforms have become a common medium for communication. Twitter is one such platform where users share their ideas, post messages which is referred as Tweet. Because of its enormous popularity, spammers have started to target the users to achieve malicious purposes. Generally, a tweet is said to be spam based on followers, following, advertising messages, URLS, hashtags. Researchers have recommended different methods to make twitter a spam-free platform. In our paper we are doing the classification based on the tweet i.e. the text they posted. We have taken a dataset from Kaggle consisting of 11969 rows and 8 attributes. Our methodology involves deep learning models, where we have used LSTM, BI LSTM, LLM and integration of the above methods. To implement the above methods, we need some python libraries NumPy, pandas, TensorFlow, OpenAI, Lang chain. Results generated by the LSTM and BILSTM are better when compared with the LLM. After the integration of LSTM and LLM, the results generated were more accurate.

**1. Introduction**

The emergence of online social network platforms, human communication has undergone a profound transformation, that has made global information sharing easier. Traditional news organizations are facing powerful competition from online social media. Most of the people are seeking to consume information from online rather than other communication medias. These combine people with their everyday lives and became very hard to escape from it. It is not only used as a communication tool, but also to share opinions, news, photos, and videos.

Among the many social media platforms, Twitter is one of the microblogging platforms, where we refer the users posts or any shared news, photos as a “Tweet” upto 280 characters. It is used by individuals, politicians, organizations and businesses to share information or opinions and engage in a real-time conversation. It has popularized hashtags usage to categorize and discover content. Twitter allows for a blue check mark for verified accounts. Over the years, Twitter has gone through many updates and changes. One of the main challenges that was faced by twitter and other platforms is spammers. Twitter has become a convenient application for spammers. They usually tweet containing URLS, mentions, spreading rumors and hashtags which direct the users to sites for their personal gain, like to spam their accounts, and steal their money.

Consequently, Twitter and many researchers have recommended techniques for spam detection to have a safe twitter platform. Additionally, twitter suspends accounts if they have low number of followers and high number of followings, fake account names, missing photo or stock photo, lack of engagement. Recently after twitter has been taken by Elon musk, when user needs blue check mark amount should be paid. The reason behind this update is for eliminating bot and imposter accounts. For classifying a tweet as spam, twitter also came up with an idea called web reputation technology for spam filtering. This method has some limitations and are unable to detect spam accounts because of the altering methods of attackers.

For finding the underlying patterns of spammer’s activities, we can use machine learning methods. After considering the previous methods of the researchers, more accurate and promising results can be generated using deep learning models. The main aim behind using the deep learning models, is they analyze data just like a human brain and they solve variety of classification problems like detecting spam tweets, fake news detection etc. They also have the ability to understand the semantics of the text behind the predictions. We are considering the LSTM, BI-LSTM and LLM as our deep learning models for the spam classification. All these models are applied on the tweet, a content-based method has been followed. The main objective is to see which models are better for classifying spam tweets. There are other different feature selection methods for classification i.e. Profile-based, Time-based, Graph-based, Automation-based.

**2. Related Work**

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J [1] The emergence of deep learning (DL) models in text categorization (TC) over classical machine learning is discussed in the paper. In task classification and sentiment analysis, text classification (TC) entails giving text labels or tags. The authors examine more than 150 DL models that have been created in the previous six years, stressing their advantages, parallels, and contributions. They also examine how well these models perform on well-known benchmarks and supply an overview of forty often used TC datasets. Applications such as question answering, spam detection, and sentiment analysis depend on text classification. Effective classification techniques are required due to the volume of text data that comes from several sources, including emails, social media, and site content. Given the growing amount of text data in industrial applications, the authors stress the significance of automatic text classification. The research divides text classification techniques into two categories: machine learning and rule-based approaches. The difficulties in gleaning knowledge from unstructured text data are emphasized. The efficacy of DL models is elucidated by a quantitative examination of their performance on benchmarks. In conclusion, the paper discusses open difficulties and makes recommendations for future approaches for text categorization research.

M. Tan, C. d. Santos, B. Xiang, and B. Zhou [2] The research supplies a deep learning framework that cuts the need for linguistic or manual feature selection in answer selection. The framework measures the similarity of questions and responses using cosine similarity, which is generated using (BILSTM) models. It is recommended to make two improvements: first, to incorporate a CNN for more detailed representations; second, to add an attention mechanism to better capture the context of the inquiry. On the TREC-QA and Insurance QA datasets, several model iterations are tested and prove a notable improvement over strong baseline. To put it simply, the authors describe a clever computer system that learns how to select the best responses on its own without being instructed on what characteristics to look for. Experiments conducted with various models prove that their strategy performs significantly better than alternative approaches. The models' performance is covered in the paper's conclusion, which proves that the CNN and BILSTM with attention combo perform better than other models on several criteria. The authors also propose to use their method in tasks such as textual entailment recognition and answer quality prediction in the future.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich [3]The article talks about GradNorm, a brand-new algorithm for effectively training deep multitask networks. Multitask networks can be thought of as brains that manage multiple tasks at once. GradNorm enhances speed and performance by aiding these networks in balancing and adapting their learning for different tasks. GradNorm offers an alternative to traditional approaches for training such networks, which presented difficulties. It smooths out the training process by dynamically adjusting the strength of the learning signals. Experiments on various task types and datasets prove that, in comparison to alternative approaches, the algorithm improves accuracy and decreases overfitting. Crucially, GradNorm makes handling many tasks easier by streamlining an earlier intricate search procedure. According to the article, this method could be used for different models that address a variety of problems in addition to multitask learning. GradNorm emphasizes the importance of gradient adjustment in obtaining better outcomes, offering a more effective and efficient method of training complex models.

Zhang, Z., Luo, P., Loy, C. C., and Tang, X [4] Their study presents a novel deep convolutional network (DCN) face detection technique. This approach achieves particularly satisfactory results on hard benchmarks such as FDDB, PASCAL Face, and AFW. It outperforms the state-of-the-art by 2.91% and reaches a high recall rate of 90.99% on FDDB. The primary breakthrough is the use of spatial structure and arrangement to score facial parts, which enables recognition even in situations with partial visibility, extreme occlusion, and position fluctuations. The suggested Faceness-Net performs better than Cascade-CNN, another face detection technique, with a 2.65% higher recall rate. Faceness-Net achieves practical runtime economy while keeping a higher recall rate, even with the fast version of Cascade-CNN being more efficient. The replacement of MCG with Edgebox for quicker object proposal and layer sharing are credited for the speed increase in Faceness-Net. The approach keeps a high recall rate even when there are fewer proposals. Although the authors recommend added modifications for higher efficiency without sacrificing detection accuracy, the runtime for VGA images on a single GPU is fifty ms. The Chinese National Natural Science Foundation supplied partial support for this study.

Yinhao Zhu, Nicholas Zabaras, PhaedonStelios Koutsourelakis, and Paris Perdikaris.[5]Their research presents an approach to model and understand physical systems described by partial differential equations (PDEs) using artificial intelligence. There is no need for labelled data because the suggested method includes the physical model's governing equations straight into the training process. Its models are effective even for high-dimensional and stochastic (random) input fields, which is a significant advance since it cuts the need for labelled data to reach high predicted accuracy. Challenges include expanding the technique to dynamic systems, strengthening the probabilistic models' dependability, merging physics-aware and scaling models to higher dimensions, and improving generalization to unknown inputs are all highlighted in the paper as being quite important. Given the circumstances; this work is a major step toward understanding PDE systems via the lens of physics-aware machine learning.

Dai, H., Liu, Z., Liao, W., Huang, X., Wu, Z., Zhao, L., Liu, W., Liu, N., Li, S., Zhu, D., et al.: Chataug [6]Enhancing natural language processing (NLP) tasks is the focus of this research, particularly for low-data scenarios. Produce more varied and labelled data for training, it presents a technique dubbed AugGPT that is based on big language models like ChatGPT. In few-shot learning challenges, AugGPT outperforms other text data augmentation techniques with encouraging results. It is not without restrictions, though, especially when it comes to medical writings where domain knowledge is essential. The authors propose more research to fine-tune general-domain models such as ChatGPT to domains. They also suggest applying AugGPT to several non-classification tasks, like text summarizing. The article emphasizes how generative models like DALLE2, and Stable Diffusion can be used to tackle computer vision issues in an analogous manner. The essay also examines the relationship between the human brain and massive language models, implying that cognitive science discoveries may improve AI capabilities. The writers hope that as these links are further investigated, fascinating discoveries will be made.

Q. Zhong, L. Ding, J. Liu, B. Du, D. Tao [7] The text talks about how ChatGPT gained notoriety for producing excellent answers to user inquiries. Although prior research has demonstrated ChatGPT's remarkable generating capabilities, there has not been much attention paid to statistically assessing its comprehension capabilities. This element is examined in the paper by comparing ChatGPT with four representative fine-tuned BERT-style models and evaluating it on the widely used GLUE benchmark. The results show that ChatGPT performs poorly on paraphrase and similarity tasks but very well on inference tasks, outperforming all BERT models by a considerable amount. Additionally, the report presents sophisticated prompting techniques that boost ChatGPT's comprehension and result in notable performance gains. Empirical studies on a range of natural language processing tasks reveal that ChatGPT excels at inference tasks but has trouble with paraphrasing and like negative examples. Despite these results, the research shows that ChatGPT can do even better understanding tasks than the potent RoBERTa-large when given sophisticated prompting tactics. ChatGPT still performs worse than the top models in various natural language comprehension tests, even though it reaches an understanding ability that is comparable to some refined BERT-style models. Overcome constraints and improve ChatGPT's comprehension performance, the paper recommends more research.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing [8] The generation and flexible understanding of coherent text has been made possible by recent developments in Natural Language Processing (NLP), converting theoretical techniques into useful applications. Large Language Models (LLMs), such as ChatGPT, have had a significant impact on copywriting and report summarizing software, among other enterprises. But there are worries that LLMs can be poisonous and show social prejudices, which could be dangerous for society and ethics. Emphasis is placed on the necessity of responsible LLM benchmarks. A qualitative investigation on ChatGPT examines several ethical issues, such as toxicity, bias, robustness, and dependability. Results point to possible ethical hazards that are not addressed by current benchmarks. The work acknowledges the limitations of empirical analysis due to unreported hyperparameters and model iterations and proposes concentrating on ethical considerations while constructing future language models. Although the study's evaluation settings may draw criticism, its main aim is to increase public awareness of ethical issues that still are unresolved in language models. The overall goal of the research is to direct future initiatives to reduce the ethical risks associated with machine applications in language models.

N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, U. Erlingsson et al [9] This study addresses a troubling problem with huge language models trained on large private datasets, such as GPT-2. The authors show that by querying these models, attackers can retrieve certain training instances. The shown attack can successfully retrieve a variety of data, such as UUIDs, code snippets, IRC discussions, and personally identifiable information. Larger language models are more vulnerable to these attacks, according to the research. The results highlight the need for measures and fixes to stop models from learning confidential information while they are being trained. The authors guess that these vulnerabilities may become more serious as language models get larger, needing the creation of methods to address privacy concerns without compromising model accuracy or training efficiency. They support more research into the mechanisms underlying model memorization, the dangers of memorization, and preventative measures.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever [10] Large language models, such as GPT-2, can learn a variety of language tasks without the need for task-specific training, according to studies. Without being explicitly taught on those tasks, the model performed well on activities like question answering, translation, and summarization when it was trained on the large WebText dataset. As the model's ability (size and complexity) increased, so did its success. Even without specialized training for language problems, the largest model, GPT-2, performed quite well. It proved its ability to generalize across several tasks by achieving top results in language modeling on various datasets. This implies a practical approach to language system construction: training huge models on broad and diverse datasets allows them to learn and perform many tasks without explicit supervision.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. [11] The study presents BERT, or Bidirectional Encoder Representations from Transformers, a novel language paradigm. In contrast to earlier models, BERT is made to be able to pre-train on unlabeled text and understand and represent language in both left and right context. This implies that BERT requires little adjustment to be perfected for a variety of tasks, such as language inference and question answering. BERT proves its simplicity and efficacy by achieving ultramodern outcomes on eleven natural language processing tasks. Most notably, it raises scores on tests such as SQuAD, MultiNLI, and GLUE. The main advantage of BERT is that it may be applied to a variety of natural language processing tasks, including resource-constrained ones, because of its ability to generalize results from deep unidirectional architectures to deep bidirectional ones.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones [12] The study describes advances in deep neural network-based character-level language modeling. While recurrent neural networks (RNNs) were previously widely used for this job, it is proved that a 64-layer transformer model with a fixed context performs noticeably better than RNNs. This model is comparable to the one that Vaswani et al. introduced in 2017. On two benchmarks, this deep transformer achieves ultramodern results: 1.06 on enwik8 and 1.13 bits per character on text8. Including auxiliary losses at intermediate network layers and sequence points is crucial for success at this level. The study also shows that while a network with twelve stacked transformer layers supplies satisfactory results, extending it to sixty-four layers improves performance even more and enables the model to grasp dependencies across large textual distances.

Conneau, A.; Schwenk, H.; Barrault, L.; and LeCun [13]Researchers have broken away from conventional techniques such as recurrent neural networks and set up a new method for NLP called VDCNN. VDCNN is a very deep convolutional neural network that, in contrast to shallow architectures, uses small convolutions and pooling operations to analyze text directly at the character level. With up to twenty-nine convolutional layers, the model's depth outperforms the state-of-the-art on several text categorization tasks. This is the first text processing application of deep convolutional networks. Operating at the character level and using a deep stack of local operations to produce a hierarchical representation of sentences are the two main tenets of the architecture. The study highlights how text and images are related, and it recommends that deeper levels of similarity be explored in future NLP models. While the study focuses on sentence categorization, further research is needed to see whether deeper convolutional encoders are useful for other sequence processing applications, such as neural machine translation.

Xiang Zhang and Yann LeCun [14] The use of temporal convolutional networks (ConvNets), a type of deep learning, to understand text from character-level inputs to abstract notions is examined in this article. The research shows how well ConvNets perform on tasks like as text categorization, sentiment analysis, and ontology classification without requiring prior knowledge of words, sentences, or syntactic structures. ConvNets may learn from scratch and do not require a word dictionary as a starting point, in contrast to earlier methods. The models prove potential for both Chinese and English cross-lingual applications. The paper makes several recommendations for future study, such as investigating transfer learning and unsupervised learning strategies and considering the use of ConvNets to tasks like named entity recognition and part-of-speech tagging. Additionally, the models supply new perspectives on symbolic systems like programming languages and mathematical equations as well as time-series data. Beyond the results that are shown, the authors think that ConvNet models may have wider ramifications for artificial intelligence.

Schmidhuber, J. [15] Deep artificial neural networks have won machine learning and pattern recognition competitions. The historical progression of deep learning is covered in this synopsis, with particular emphasis on evolutionary computation, reinforcement learning, unsupervised learning, and supervised learning. Deep and shallow learners are distinguished by the depth of credit assignment paths, which influences their ability to relate causes to consequences. Numerous fields, including segmentation, object detection, and picture recognition, have used deep learning. Brain-like RNNs could improve by activating only necessary units and minimizing communication costs. Additionally, the text delves into issues such as neural network computation costs and credit assignment paths. Neural networks may in the future be able to actively recognize patterns, weigh energy costs, and efficiently adopt brain-like structures. The aim might be self-optimizing general-purpose learning algorithms, however there is still work to be done in terms of actual application.

I Ahmad, M Yousaf, S Yousaf, and M O Ahmad [16] Spread of knowledge has been altered by the internet and social media platforms like Facebook and Twitter. Although people share a lot, not all of the information is correct or true. Even for specialists, it can be challenging to decide whether a story is true or not. We propose to aid with machine learning. Various textual elements that can show whether news is bogus are examined in our study. They evaluate a variety of machine learning techniques using actual data. Our approach outperforms stand-alone techniques. It is difficult to find bogus news. We must investigate the factors that contribute to its spread and find sources using machines and graphs. Another concept for the future is to search for fake news in real-time videos. There is still a lot to solve in stopping fake news from spreading.

Ni, B., Guo, Z., Li, J., & Jiang, M [17] The emergence of social media sites on the internet has raised awareness of bogus news. This was investigated in a research project that looked at what makes it difficult to find fake news. Propensity Score Matching (PSM) was employed to choose features that are effective even when other factors cause interference. The findings proved that PSM improved the prediction of fake news significantly above word frequency alone in fake news detection algorithms. They experimented with several techniques, such as random forests and logistic regression, and PSM helped them all get better. Put another way, they discovered a more effective way to decide whether online news is bogus by employing clever techniques to find the key indicators.

Khan, J. Y., Khondaker, M., Islam, T., Iqbal, A., & Afroz [18] The study investigates the issue of false information spreading on social media and the difficulties in finding it. Assess the performance of nineteen machine learning techniques, researchers ran tests on three different datasets. Traditional, deep learning, and sophisticated pre-trained language models like BERT are some of these models. Notably, BERT-based models performed better than other models on all datasets, with small datasets showing particular strengths. Neural networks with a large enough dataset produced outcomes comparable to those of naive Bayes with n-gram. The length of the dataset and the information contained in news stories affect how effective LSTM-based models are; these models have a better chance of succeeding when there is sufficient information available. The results imply that pre-trained algorithms such as BERT are better at finding bogus news, particularly in languages with low levels of electronic content. The study intends to direct future research and aid entities, including news portals and social media platforms, in selecting proper models for the identification of fake news. Future efforts will concentrate on combating misinformation and fake news connected to health during the COVID-19 epidemic.

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin [19] The authors introduce BERT, a model, to classify documents, even though it may seem unconventional due to certain task characteristics. They prove that a simple classification model using BERT achieves ultramodern results on four datasets, despite concerns about the model's suitability for the task. Mitigate computational costs, they distill knowledge from the large BERT model to smaller bidirectional LSTMs, achieving similar performance with significantly fewer parameters. The main contribution of the paper is the enhancement of baselines for document classification. They fine-tune BERT and use its knowledge to improve a lightweight BiLSTM model, achieving comparable performance with substantial parameter reduction and faster inference times. The authors suggest further exploration into the effects of distillation across various neural network architectures and propose investigating model compression techniques for transformer models in future research.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin [20] Complex neural networks, such as BERT, ELMo, and GPT, have gained attention in the field of language processing. People thought that the more primitive, older networks used to understand language were no longer relevant. A recent study casts doubt on this theory, though. The researchers prove that without changing their structure, incorporating added characteristics, or adding more data, simple, lightweight neural networks may still be competitive. They suggest condensing BERT information into a more straightforward BiLSTM model. While using far fewer parameters and requiring less time for inference, this condensed model performs comparably to ELMo. The results imply that simpler BiLSTMs might do language tasks more successfully than previously believed. Additionally, the findings suggests that future research will examine amazingly simple or slightly more complicated structures, ranging from simple convolutional neural networks to complex models including attention mechanisms and word interactions.

**3. Data**

We have considered a dataset from a website called Kaggle. Our dataset contains 8 attributes and 11968 rows. Below is the list of attributes:

Tweet - Text that was tweeted by the users

following - Number of people the account that tweeted is following

followers – Number of people following the account that tweeted

actions – Total Number of favorites, replies and retweets of said tweet

is\_retweet – if 0 its not retweet, if 1 it is a retweet location – self written location provided by the user

Type – Either Quality or spam

As we are following content-based method, only some attributes i.e. Tweet and type are considered from the dataset for our predictions.

A screenshot of a computer

Description automatically generated

**Fig 3.1 Dataset**

**4. Methodology**

The first step of any model is data preprocessing, an essential step for cleaning and transforming raw data for analysis. For our dataset also, we have done some preprocessing steps like checking and removing null values, duplicates elimination. Using some Natural Language Processing (NLP) techniques we have removed the stop words in the tweet column, converting the tweet into lower case and also replaced the characters other than alphabets to space. There are a variety of features that are useful for spam detection, but not all of them are helpful. In our dataset also, we are considering only some features for our content-based method, the other features are ignored. After the preprocessing we are creating a new dataset with Tweet and Type as our columns where the 2 labels Quality and spam equals to 0 and 1 respectively. The next step for our analysis is splitting the dataset into training and testing sets with 80 and 20. We cannot directly give a text sample for the model. For this purpose we used a method text\_to\_sequences to convert into sequence of integers. The next step is padding the sequence, as LSTM only accepts data with equal length. We have used two LSTM layers, one with the 128 units and the other with 256 units with sigmoid as activation function. The epoch number in the model decides how many times it should run on the data and generates the best accuracy. 77% and 79% are the respective accuracies for the two models. The same we have done for the Bi-LSTM, where we got 79% and 76% accuracies.

Our next approach is using Large Language models. Of many models that are available publicly, GPT 3.5 has more popularity. To see the model’s behavior on our dataset, we are using prompt engineering techniques. GPT 3.5 model is not applied on the whole dataset, we are using only some of the dataset. Using a technique to get the sample of the data which best represents the population called stratified sampling. This is how we obtained our sample, and we took into account 100 samples from the dataset. Among them only test data is used for our GPT 3.5 model, as the model is already pre-trained, we don’t need to apply it on the training data. For applying the prompt engineering methods on our data, we need to give a template to make it more interactive. When instantiating the AI, we have tried different values for the temperature parameter. Using the Open AI API with 3 prompting techniques zero-shot, one-shot and few-shot learning we have built our model and achieved an accuracy of 52%, 52% and 51%. For the three learning techniques, there are some templates or elements which make a good prompt and to improve the interaction with the LLM model. The template consists of Context, Note, Task, Instruction and explanation which is optional. We are providing a context to it as “you are a spam detection system for a social media platform and your goal is to classify as spam or not”. Two tasks are provided to it, one is to classify, and the other is describe the reason for the label in less than 3 sentences. In the instruction we are specifying how the output should be i.e. a label and the reason for the label. For the one shot and few shot we are providing some examples for training as they require for predicting. Input for these templates is a tweet. Now for the same stratified sample, we applied LSTM and BILSTM models on the training and testing set, where the accuracies are 64% and 60%.

From the above observation, by comparing both the approaches we can say that LSTM and BILSTM are performing better with more accuracy. The reason behind this is, as the prompting techniques does not receive the whole data for training, they know only some of the examples, due to this they achieved very low results compared to other models. So, we came with an approach of integrating LSTM with LLM model as LSTM is more accurate in providing the output label and LLM was more accurate in giving correct explanation. We are taking the LSTM label and giving to the LLM as input along with the Tweet and trying to get the reason for the label from the GPT 3.5. When we are building this model, we tried with only few examples, as giving whole data it takes a lot of time for the GPT 3.5. A tweet was preprocessed and converted to sequences along with the padding. Now we have imported the LSTM model with high accuracy i.e. 256 layers and passing the processed text to the model. After the model has predicted, we are using a commonly used activation function called sigmoid to decide spam or not. If the predicted value is greater than 0.5, then we are considering the label as “Spam” otherwise the label is “non-spam”. Using this integration approach, overall performance was improved and an accurate label along with the reason was generated.

**4.1 LSTM**

LSTM stands for Long Short-Term Memory. It is a deep learning neural network which has the capability to information persistence. A special type of Recurrent Neural network (RNN) is used to handle sequential data such as time series, speech and text data. It was designed by Hochreiter and Schmidhuber to address the problems faced by RNN and other machine learning algorithms. LSTMS excels in capturing long-term dependencies using its memory cell, which is used to store information for longer time and handles the vanishing gradient problem faced by RNN. It works well for capturing complex issues like speech recognition and machine translation where they require to understand order dependence. For these purposes LSTM works better than any other deep learning method. The reason behind the effective working and understanding of sequential data is they use feedback connections. Usually, complete data sequences are interpreted instead of single data points. In LSTM, each word in a sentence is considered as token, it retains useful information about the previous word to aid in processing new data or output the label.

Considering an example Betty eats Samosa every day, her favorite cuisine is Indian (it needs to predict the word Indian). From the above sentence we can say that, based on the training data it understands the meaning of the before words in a sentence and gives the output as Indian. LSTM decides which word to discard or retain from its memory. Another example is Betty eats samosa every day, it is an Indian cuisine, she also eats pasta and cheese, it is an Italian cuisine. The model decides by itself and gives importance to the next word pasta stores it and discard the information of samosa. If it wants to retain it can hold for a longer time. The long-term dependencies for the LSTM can be done using a memory cell which helps to maintain a stable flow of gradient in backpropagation. In general, all the traditional deep learning methods along with RNN have the gradient issue problem. The weights computed for this gradient can become small during the backpropagation, because they are multiplied through many time stamps which is making it hard for the long-term dependencies. This memory cell of LSTM which helps in solving many complex problems is controlled by 3 gates: input gate, output gate and forget gate. Every gate has some function, where the first gate decides if the information received from the preceding timestamp to be remembered or irrelevant and can be ignored, the second gate try to learn or update new information from the input that received, and the third gate gives the output of the updated information to next timestamp. A single time step that equals with the one cycle of LSTM. Based on the requirement we can use n\_lstm layers for prediction.

A diagram of a new information

Description automatically generated

**Fig 4.1.1 Structure of LSTM Cell**

The above figure is the architecture of LSTM. Cell state is known as the Long-term memory and hidden state as Short-term memory. In the above the cell state resembles a conveyor belt, which flows straight down the entire chain. Information can just pass through it unaltered. Unlike RNN, it does not have a simple repeating module structure, it contains four which are interacting in a special way. In all the LSTM time-steps, the first stage is to decide which data to discard from its cell state by using the sigmoid activation layer called as “Forget gate”. It takes the current input and preceding hidden state, outputs the value between 0 and 1 for each number in the cell state. It forgets everything if the value is 0 and remembers all the information then the value is 1. All this is decided by the forget layer. We will store new information in the cell state, this is decided by the input gate layer and the tanh layer which creates a new value to output to the next state. Combining these two for updating the state is the next step. We forget the things we decided before and update each state. Finally, a output will be given which is to decided based on the current cell state. The output will be a bit different version. A diagram of a diagram

Description automatically generated

**Fig 4.1.2 LSTM Architecture**

Even though LSTM is a special type of RNN, it has more advanced features than the other deep learning models. In sequential data, RNN can only be trained in forward direction with a no long-term dependency learning and cannot handle vanishing gradient problem occurring during the training. These problems of RNN, make it harder to capture information and persist it. That’s why LSTM was developed as it can solve all the problems of RNN and can understand the importance of context, even though there is a significant time gap between events. When a model requires understanding the context, LSTM comes into picture.

There are many applications of LSTM like language modeling to generate grammatically correct sentences between words in a sentence, speech recognition for finding patterns in a speech and matching to the correct text, Time series forecasting, for the predictions of future events it learns the patterns in data. If we want to predict sales of a shop monthly, we see the whole year pattern and store the previous year prediction understands it and gives the output we need. LSTMs can be combined with other neural network architecture like Convolutional neural network for analyzing video such as object detection to extract information from it.

**4.2 BI-LSTM**

It stands for Bidirectional long short-term memory is one of the deep learning model, another special type of Recurrent Neural network(RNN) but it is different from the traditional LSTM. In the LSTM the information flows in only forward direction it makes predictions based on the understanding of the previous word. Certain patterns cannot be captured by the LSTM. These limitations are addressed by the BI-LSTM. It is useful in complex patterns and in tasks when the whole context of data is important both the past and future for capturing or accessing the information in both ways simultaneously. The Bilstm process input in both forward and backward directions using two LSTM layers. One layer is for processing the data in forward direction and the layer is for processing in another direction.

Consider an example, Ben loves apple, it keeps him healthy. Until the word apple it understands that as company using LSTM, it does not consider the future and past words for the prediction. To understand the whole concept properly, it needs future word also, here the future word is healthy. Using these future and past words it predicts that the word as fruit.

A diagram of a block diagram

Description automatically generated

**Fig 4.2.1 BILSTM Architecture**

From the above architecture we can interpret that, there are two lstm networks, one LSTM layer gets the sequence of token in forward direction i.e. left to right in a sentence and the other lstm layer is used to read from right to left. The output from the two LSTM layers are aggregated using methods like concatenation, sum and average. Gradient calculations need not be done manually, these frameworks have built-in mechanisms to handle the models. For properly integrating the gradients in both the passes, in the training process the bilstm adds some complexity. It used in various applications like sentiment analysis, named entity recognition and for the POS tagging.

**4.3 Large Language Model (LLM)**

Using a powerful Artificial Intelligence algorithms model was developed to understand the human language, this model is called as Large Language model. These models are trained on huge amount of data to perform various natural language processing tasks and also to capture more complex patterns. The most common architecture of LLM is transformer networks, which consists of neural networks with encoder and decoder. This has a capability for self-attention. It just works like a human brain, where it interprets and generates the text. It is capable of unsupervised training and when they learn on huge data they acquire fundamental language, grammar and knowledge. These process data not in a sequential process they process in parallel which consequently cuts down the training time. During the training of the LLM trillions of parameters are collected which helps to capture patterns.

There are many evolutions on these language models where it started with specific task helper to general-purpose task solver. Each model has its own capabilites, but Large Language model has many more features when compared with the before evolutions and capabilities of LLMs have increased the performance of models. They have many abilities which makes the AI algorithms more powerful than the other LLM models.

A diagram of a task

Description automatically generated

**Fig 4.3.1 Evolution of LLM**

In the traditional machine learning algorithms, to represent each word in a sentence into a numerical table. Word relationships such as with similar meanings were not able to recognize in the above method. LLM will resolve this limitation by using word embeddings which is also referred as multi-dimensional vectors. In the vector space words with same meanings are kept close to each other. The way LLMs represent words is a crucial component of how they function. Using the transformer architecture, encoder converts the text to numerical representations with the word embeddings and interprets the relationship between words including the parts of speech and the context of words. We can apply the LLMs through the decoder only after the numerical representation for generating a unique output. The LLMs have self-learning techniques where it adjusts the parameter values iteratively until the next token was predicted correctly from the preceding input token sequence. For increasing the likelihood it optimizes the parameters (weight and bias) for next tokens in the data.

OpenAI GPT-3 model is one of the examples of LLM model, which can identify patterns and generate a relevant text taking account into variety of topics and tasks. It is flexible and can be applied for various tasks in natural language processing. We can access this Open AI using the API provided and allowing users to integrate its capabilities. Open AI takes two parameters one is the temperature where the values ranges from 0 to 1 and the other one is the API\_key. For the temperature parameter, if we set it to 0, the GPT-3.5 does not think in a broad way and gives answers directly, if the value is set to 1, then the model thinks in a Broadway by considering all the things just like the human brain and generates the output. There are many other LLMs: palm2, t5 by google and Llama by meta. Compare to other models, GPT-3.5 is more flexible as the other algorithms does not generate a descriptive text and are less used by the developers. It requires only minimal training to accomplish its tasks and uses prompt engineering for this purpose.. These LLM models need to be pre-trained and finetuned to solve the complex use cases. A human intervention is necessary to make some models, such as ChatGPT, less harmful to users through the application of reinforcement learning and human feedback.

A diagram of a diagram

Description automatically generated

**Fig 4.3.2 Use cases**

**4.4 Prompt Engineering**

The process of giving or designing inputs to a AI generative tools for producing optimal outputs is prompt engineering. In simple words, how effectively we are interacting with an AI to see if the desired outcomes are producing or not. These all desired outcomes depends on the right prompt, it might misunderstand your request, if the prompt is not given correctly.

Lets take an example of google search engine, we need to give inputs correctly so it interprets rightly and gives the outcome. If we want to search for a keyword in the pages, we provide a sentence with a quotation mark for the word we want. Google generates the links containing that word. Other example is Consider asking for slides on some particular topic in .ppt or .pdf, the results generated are only those formats. All these are ways of giving prompts to the AI tools for getting optimized results. More effective prompts can be given if the relationship between the parameters and model outputs are more comprehending. In today’s world, there are many evolutions on the LLMs, this is where the prompt engineering matters as every model gives a right answer, but the main concept is assuring AI comprehends the context and the purpose of each query, also we should see that every word in a prompt is important, a small difference in the sentence can change the whole generated text from an AI. Every prompt has a template which consists of Context, Instruction, Input data and output format. When the model data is increasing, then designing effective prompts will be more challenging. There are also some considerations while creating a prompt: the information we provide should be very brief, open-ended questions and constraints should be kept highly so it can understand better the requirements.

Techniques for prompt engineering:

A) Zero-shot learning: A machine learning technique, which uses transfer learning mechanism using the instances that are already fed while training. In the supervised learning, we will be training the model on some specific set of classes and if we give the classes which are not present in the training data, it might be difficult to classify for the model. These types of limitations can be solved by zero-shot learning, where it make predictions based on the unseen data which were not given during training. It just works like a human brain, if we someone says a zebra just looks like an horse but with white and black stripes. Now they will easily recognize it if they see zebra picture.

A screenshot of a computer

Description automatically generated

**Fig 4.4.1 Example of Zero-shot learning**

The above picture is an example of the zero-shot learning. In this we are not giving any training samples for the AI, just instructions, input and asking for output.

B) One-shot Learning: This type of machine learning algorithm requires only one example for each category to predict the output, mostly required in deep learning models. It has an architecture called Siamese networks which can learn the similarity between the instances and their relationship. It is mainly used in voice cloning, image recognition and other applications in medicine.

A screenshot of a phone

Description automatically generated

**Fig 4.4.2 Example of one-shot learning**

In the above example, we are giving one training example for the model to generate the output based on the example given.

C) Few-shot Learning: Instead of feeding whole data for training the model, in this only few examples are given to generate an output. It is same as meta-learning shot; in the training phase it is generated with several examples and categorizes well on unseen data. This learning reduces the computational cost and in data collection also. When there is insufficient data, this model helps a lot and used to classify anomalies.

A screenshot of a group

Description automatically generated

**Fig 4.4.3 Example of few-shot learning**

For this we are providing a set of examples during training for the model, to predict the required result. It uses the same architecture as one-shot learning which is often used for learning similarity.

**5. Results**

We performed analysis on entire data different combinations of LSTM and BI-LSTM layers. We have taken out 100 samples from the using stratified sampling method. For this sample we have taken out the training and testing sets, applied the LLM model on the testing set and LSTM,BILSTM on the 100 samples. Then we integrated LSTM with the LLM.

**5.1 LSTM Results:**

Number of LSTM Layers = 128

**Training results:**

A screenshot of a computer screen

Description automatically generated

**Fig 5.1.1 Performance metrics on Training Data**

A yellow and purple squares with black numbers

Description automatically generated

**Fig 5.1.2 Confusion matrix on Training Data**

**Testing Results:**

A screenshot of a computer code

Description automatically generated

**Fig 5.1.3 Performance metrics on Testing Data**

A yellow and purple squares with numbers

Description automatically generated

**Fig 5.1.4 Confusion matrix on Testing Data**

**5.2 LSTM Results:**

Number of LSTM Layers = 256

**Training Results:**

A screenshot of a computer screen

Description automatically generated

**Fig 5.2.1 Performance metrics on Training Data**

A yellow and purple squares with numbers

Description automatically generated

**Fig 5.2.2 Confusion matrix on Training Data**

**Testing Results:**

**A screenshot of a computer screen

Description automatically generated**

**Fig 5.2.3 Performance metrics on Testing Data**

**A chart of spam

Description automatically generated**

**Fig 5.2.4 Confusion matrix on Testing Data**

**5.3 BI-LSTM Results:**

Number of BILSTM layers=128

**Training Results:**

**A screenshot of a computer screen

Description automatically generated**

**Fig 5.3.1 Performance metrics on Training Data**

**A screenshot of a computer screen

Description automatically generated**

**Fig 5.3.2 Confusion matrix on Training Data**

**Testing Results:**

**A screenshot of a computer

Description automatically generated**

**Fig 5.3.3 Performance metrics on Testing Data**

**A yellow and purple squares with numbers

Description automatically generated**

**Fig 5.3.4 Confusion matrix on Testing Data**

**5.4 BI-LSTM Results:**

Number of BI-LSTM layers=256

**Training Results:**

**A screenshot of a computer

Description automatically generated**

**Fig 5.4.1 Performance metrics on Training Data**

**A yellow and purple squares with numbers

Description automatically generated**

**Fig 5.4.2 Confusion matrix on Training Data**

**Testing Results:**

**A number of numbers on a white background

Description automatically generated**

**Fig 5.4.3 Performance metrics on Testing Data**

**A screenshot of a computer screen

Description automatically generated**

**Fig 5.4.4 Confusion matrix on Testing Data**

**5.5 LLM Results:**

**A close-up of a screen

Description automatically generated**

**Fig 5.5 LLM Results**

**5.5.1 Zero-shot Results:**

**A screenshot of a computer

Description automatically generated**

**Fig 5.5.1.1 Performance metrics**

**A chart with numbers and labels

Description automatically generated with medium confidence**

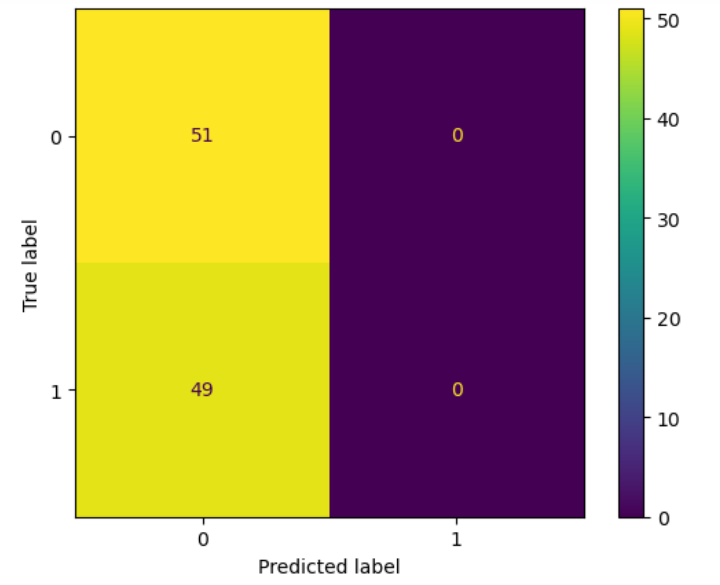
**Fig 5.5.1.2 Confusion matrix of Zero-shot**

**5.5.2 One-shot Results:**

**A screenshot of a computer

Description automatically generated**

**Fig 5.5.2.1 Performance metrics**

****

**Fig 5.5.2.2 Confusion matrix of One-shot**

**5.5.3 Few-shot Results:**

**A screenshot of a computer screen

Description automatically generated**

**Fig 5.5.3.1 Performance metrics**

**A yellow and purple rectangular boxes with numbers

Description automatically generated**

**Fig 5.5.3.2 Confusion matrix of Few-shot**

**5.6 Integrated Results:**

A screen shot of a message

Description automatically generated

**Fig 5.6.1 Results of Integrated model**

The above figure shows the results of Integration of LSTM and LLM models. For the same sentence, before integrating we applied the LLM model, and it predicted the label as spam but the ground truth is non spam. From the results achieved in the above figure we can say that the integration outperforms than the other individual models.

A screen shot of a computer

Description automatically generated

**Fig 5.6.2 Results of LLM modelsss**

**6. Discussion**

For our analysis of spam tweets, we have taken a Twitter tweet dataset from Kaggle website. In this dataset only the tweet and type columns are considered, and other columns are ignored. Using preprocessing methods, we have cleaned the dataset so it can be used as input to the models. Two layers, i.e. 128 and 256 of LSTM and BILSTM are used for our model and then we applied our LLM on the testing data which was taken from the 100 samples of the whole dataset. On this we have applied our deep learning models LSTM and BILSTM, where the accuracy for LSTM and BILSTM are more when compared with the LLM using different architectures (zero-shot, one-shot and few-shot). So, we have another model which is integrating of two models, where we are taking the label from the LSTM model which has more accuracy and asking our LLM model reason for that label. This model improved the overall performance of the model.

**7. Conclusion and Future Work**

Using Deep Learning methods, LSTM, BILSTM and LLM we have developed a model for classifying twitter tweet spam information. LSTM and BILSTM with 256 layers show higher accuracy when compared with the 128 layers models. Also the LLM model using GPT-3.5 shows less accuracy than the other models because these are trained on few examples. The integration of two models leverages the overall performance. LSTM gives the correct label and our OpenAI was more accurate in giving the explanation. There are many benefits of integrating, a more robust solutions are produced by the models. LLM model has some limitations like less examples for the training which makes it difficult to categorize the unseen classes. Also, there is one more limitation of the LLM i.e. the number of tokens are limited for our GPT-3.5. These challenges can be considered for future work. In our model we have used the content-based method for classifying tweets, in further we can use hybrid models not only the text but also the other attributes. In future the spam tweets might be in the form of image, video or audio and we should be able to sort it from the real tweets.

**8. References**

[1] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. 2022. Deep learning–based text classification. ACM Com-putting Surveys, 54(3):1–40.

[2] M. Tan, C. d. Santos, B. Xiang, and B. Zhou, “Lstm-based deep learning models for non-factoid answer selection,” arXiv preprint arXiv:1511.04108, 2015.

[3] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2017. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks pages 1–10.

[4] Zhang, Z., Luo, P., Loy, C. C., and Tang, X. Facial landmark detection by deep multi-task learning. In European Conference on Computer Vision, pp. 94–108. Springer, 2014.

[5] Yinhao Zhu, Nicholas Zabaras, PhaedonStelios Koutsourelakis, and Paris Perdikaris. Physicsconstrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labelled data. Journal of Computational Physics, 394:56–81, 2019

[6] Dai, H., Liu, Z., Liao, W., Huang, X., Wu, Z., Zhao, L., Liu, W., Liu, N., Li, S., Zhu, D., et al.: Chataug: Leveraging chatgpt for text data augmentation. arXiv preprint arXiv:2302.13007 (2023)

[7] Q. Zhong, L. Ding, J. Liu, B. Du, D. Tao, Can Chatgpt Understand Too? a Comparative Study on Chatgpt and Fine-Tuned Bert, 2023 arXiv preprint arXiv: 2302.10198.

[8] Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. 2023. Exploring ai ethics of chatgpt: A diagnostic analysis. arXiv preprint.

[9] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, U. Erlingsson et al., “Extracting training data from large language models.” in USENIX Security Symposium, vol. 6, 2021.

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” 2019.

[11] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[12] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2018. Character-level language modeling with deeper self-attention. arXiv preprint arXiv:1808.04444.

[13] Conneau, A.; Schwenk, H.; Barrault, L.; and LeCun, Y. 2016. Very deep convolutional networks for natural language processing. CoRR abs/1606.01781.

[14] Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. arXiv preprint arXiv:1502.01710.

[15] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. [https://doi.org/10.1016/j.neunet. 2014.09.003](https://doi.org/10.1016/j.neunet.%202014.09.003)

[16] I Ahmad, M Yousaf, S Yousaf, and M O Ahmad. Fake news detection using machine learning ensemble methods. Complexity, 2020, 2020. cited By 0.

[17] Ni, B., Guo, Z., Li, J., & Jiang, M. (2020). Improving Generalizability of Fake News Detection Methods using Propensity Score Matching. Social and Information Networks. <https://arxiv.org/abs/2002.00838>

[18] Khan, J. Y., Khondaker, M., Islam, T., Iqbal, A., & Afroz, S. (2019). A benchmark study on machine learning methods for fake news detection. Computation and Language. <https://arxiv.org/abs/1905.04749>

[19] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: Bert for document classification. arXiv preprint arXiv:1904.08398, 2019.

[20] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task specific knowledge from BERT into simple neural networks. arxiv/1903.12136.