A

Major Project

On

HIGH PERFORMANCE ARTICLE WRITING USING DEEP LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Paridhi Wahii(187R1A0530)

Abisha Winslet(187R1A0508)

Suchith Adepu(187R1A0555)

Under the Guidance of

Dr.T.S. MASTAN RAO

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**HIGH PERFORMANCE ARTICLE WRITING USING DEEP LEARNING**" being submitted **by PARIDHI WAHII(187R1A0530), ABISHA WINSLET BERI(187R1A0508) & SUCHITH ADEPU(187R1A0555)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. T. S. Mastan Rao
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on___________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide, **Dr. T. S. Mastan Rao** ,Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work . The blessing , help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Mr. A . Uday Kiran, Mr. J. Narasimharao, Dr. T. S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar,** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion ofthis project.

PARIDHI WAHII(187R1A0530)

ABISHA WINSLET(187R1A0508)

SUCHITH ADEPU(187R1A0555)

ABSTRACT

Content writers spend endless nights creating content which might or might not aid their requirement. A lot of times the content produced is not good in terms of qualitative nature of the entire text. Technical writing can be equally challenging The challenge of understanding topics, finding data specific to the need of topic, making sure information acquired is validated and finding time to do the research required can be cumbersome The model we propose aims to shorten the time it takes to create high-quality content. Traditional supervised state-of-the-art models, which were utilized in content generation, are based on a variety of assumptions from various research viewpoints but it produced unwanted data which lead to decelerating efficiency. Long strings of text were generated, but they lacked creativity and human sentiment, and they couldn't tell the difference between good content and crude language. The content created was often repetitive in nature. Our project aims to help in generating content for technical articles which require factual information or just reference material for writers needing good quality data in a short amount of time. We use a transformer-decoder only model also known as GPT-2, which is an unsupervised language model. It showed that training on larger dataset and having more parameters improved the capability of language model to understand tasks and surpass the supervised language model. As a language model it produces large amounts of quality copy. Through our project content can be enhanced and produced faster, while humans can intervene to insert creativity and uniqueness. Deep learning and natural language processing are the two key features of artificial intelligence (AI) technologies that help AI tools in using prevailing information about any subject from the database and processing it accordingly.

LIST OF FIGURES

LIST OF TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENT

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The proposed system is titled as "High Performance Article Writer Using Deep Learning" which is a deep learning-based system that is capable of automatically writing articles on a variety of topics. The model is trained on a dataset that comprises of more than six thousand abstracts of different research papers. It is trained to follow a formal tone and without any grammatical errors for text generation and can write articles for business blogs, technical writing, websites, corporate communication, research papers, newspaper articles etc.

1.2 PROJECT PURPOSE

The content generation models are revolutionizing the domain by generating human-like text. It has gained wide popularity recently in many domains like news, social networks, movie scriptwriting, and poetry composition, to name a few. The proposed system takes the title (keyword or string of keywords) and the length as input and produces relevant output .Text generated can be used by the user to rework or just as reference material to refer . It aids in the preparation of project papers, journalistic articles, or simply reference material for authors experiencing writers block. The level of creativity can be decided by user as per their preference, if they want content to be entirely factual or let the system take creative liberty . It's takes factual information and presents in a digestible form — relevant to the audience. It aims to save time and effort by producing large quantity of text in a short amount of time while also being able to mimic human sentiment. The text produced is factually correct as the system is trained on data taken from validated sources . As technical articles, online publications or any writing task requiring large amounts of text , require research and time the system can be beneficial to people who have to write lengthy articles or require reference material with the generated text holding human sentiment and following grammar constraints of the English language.

1.3 PROJECT FEATURES

- The proposed system provides authentic and validated information to the user.
- It is trained to be grammatically correct
- Understand the tone of technical writing.
- It provides quality and relevant content
- Content generated takes less time.
- Reduces effort of writer.
- It produces content that is specific to the title taken in as input.
- The length and degree of creativity can be determined by user.
- Generated content is easily accessible to user in .txt format.
- User friendly
- Bulk text is generated in short amount of time

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

System Analysis is the important phase in the system development process. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.2 PROBLEM DEFINITION

Generating coherent long text is useful in myriad applications of creating reports, essays, and other long-form content. While humans have traditionally curated content on their own, the process is grueling as it takes a lot of time and effort to be able to create one good quality work. The process requires lots of research, hours to correct language and validate facts(if working on a more formal form of writing task) or sometimes years if one has to publish creative work which requires a lot of ideas and reference material. Technology has come a long way and there have been multiple attempts to produce bulk text which holds relevancy and sentences being coherent . The problem is particularly challenging as it demands models to capture global context, plan content, and produce local words in a consistent manner. There is a lack of extensive review and an up-to-date body of knowledge of text generation deep learning models. Topic specific content is hard to accumulate especially if required to write factual articles or research papers. The content required should be validated and authentic. A lot of research goes into writing a technical paper which also requires a lot of time and effort.

Good writing skills also play an important role to capture interest of the audience and to be able to explain the relevancy and need of the researched project when it comes down to writing a research paper. A lot of the time, the content is poor in terms of the overall quality of the language or fails to capture interest of the targeted audience. The importance of language modeling comes from its involvement in many language processing tasks such as conversational system, speech to text, and text summarization. The language models are typically trained to learn the occurrence of next word in a sequence based on previous words in the text. However, when it comes to testing, it is highly expected that the entire sequence will be generated from the scratch which is computationally not suitable to many applications.

2.3 EXISTING SYSTEM

Language models estimate the probability of words appearing in a sentence, or of the sentence itself existing. Stochastic models like the Markov Chain is one of the first text creation algorithms (based on a random distribution of probabilities). The model has no memory; therefore, the prediction is just based on the variable's current state (it forgets the past states; it is independent of preceding states. LSTM (Long Short-Term Memory) neural networks have a long short-term memory. They learn specific patterns, making them a good tool for modelling sequential data. Local coherence is there in the created text, but it lacks logic and often require a burdensome amount of training data to be useful for specific tasks and domains. Massive deep learning language models are designed to tackle these pervasive training data issues. Recurrent neural networks were introduced thirty years ago. The traditional neural network model is mainly based on the feedforward neural network. The input dimension information of the feedforward neural network is independent, and the dimension of the input information is fixed. In natural language processing tasks, each word depends on the previously entered words. Natural language input is a sequence of information, the length of the sequence is variable, and there is a connection between the sequence information.

Therefore, the feedforward neural network is restricted in natural language processing tasks models are useful for sequenced tasks such as abstractive summarization, machine translation and general natural language generation. They process words sequentially, one word at a time. As a result, these models are hard to parallelize and poor at retaining contextual relationships across long text inputs.

2.3.1. LIMITATIONS OF EXISTING SYSTEM

- Content produce by different NLP Languages or language models is repetitive.
- Errors in text or speech.
- Colloquialisms and slang.
- Domain-specific language.
- Low-resource languages.

2.4 PROPOSED SYSTEM

To tackle the problem of bulk generation of creative text that can be used for various writing tasks we proposed a language model that uses the simple GPT-2 AI. The proposed system has been trained on a custom dataset which includes abstracts of over 6000 research papers. The dataset was created using the python research-Insights library and the scrapy framework . A web crawler was made to crawl validated education websites and thousands of research papers to train the system to be able to write formal writing works. The dataset amount to 70 mb and was stored on the google drive .To train the model , we used google colab and mounted the drive and used it as our primary storage to store files. We used the simple GPT-2 model which is a transformer-decoder only model which employs two layers namely the masked self-attention and the feed-forward neural network. The main aim was to build a content writer which can produce high-quality content while also being creative. The proposed system can be useful to develop content for readers on a daily basis. It generates articles based on a snippet of text, a sentence or even a passage from an article given as input by the user. It allows the user to define the length of text and degree of innovation in the text to be produced. We employ a transformer-decoder language model also known as GPT-2 which has 124 million parameters. The model is fine-tuned till desired results were achieved in the produced content.

The recommended temperature lies.between 0.7 to 0.9 which produced good results. The content generated can be used for a variety of writing tasks or just as reference material and can be downloaded in a .txt format file.

2.4.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Writing news articles-given only a title
- Reduces human effort
- Able to mimic human sentiment
- Greater efficiency
- Better quality control
- Can produce bulk text in less time.

2.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.5.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminaryinvestigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.

- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the system is economically possible for development.

2.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement; as onlyminimal or null changes are required for implementing this system.

2.5.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQURIMENTS

For developing the application, the following are the Hardware Requirements:

- System : Intel i5
- Hard Disk : 1 TB
- Ram : 8GB
- GPU : Nvidia T4

2.6.2 SOFTWARE REQURIMENTS:

For developing the application, the following are the Software Requirements:

- Operating System : Window 7,8,10
- Backend Language : Python
- Tool : Jupyter Notebook, Google Colab

3. LITERATURE REVIEW

3.1 LITERATURE SURVEY

In [1], The paper uses two popular AI models for text generation and perform a comparative analysis. OpenAI GPT-2 and BERT models are used for prediction and generation of text .GPT-2 is trained on two corpora to generate long sentences along with articles while the BERT model is used for prediction of immediate words on the basis of given context .

In[2],The paper explores the use of pre-trained NLP models, BERT and OpenAI GPT-2 to gather coronavirus related information and literature. Text summarization is performed on the COVID-19 Open Research Dataset (CORD-19). The models are used to perform mapping from a specific keyword selected to generate summary text, resulting in an abstract summary. The goal is to bring researchers closer to fast growing publications of COVID literature and uses the ROUGE metric to evaluate the text summarization.

In [3], Generative Pretrained Transformer 2 (GPT-2), an NLP model is used to experiment with text generation to test if fake reviews generated by AI tools for academic purposes can be feasible .The 355M version of GPT-2is used and fine tuned with a corpus of review reports which is based on PeerRead dataset.

In[4],The paper observes that a natural language model can perform down stream tasks in a zero shot setting without any sort of modification and when a large language model is trained on a large and diverse dataset , the model's performance increases across many domains. It is demonstrated that the language models when experimented with a large dataset begin to learn tasks without any explicit supervision. The resulting model trained on the CoQA dataset – excels in comparison with other baseline models

In[5], The paper explores way to control attributes of the generation capabilities of different transformer based language models that have been trained on large The proposed model combines a pretrained language model with simple classifiers .The attributes to control generation are use a bag of words (BoW) which relates to a topic and a linear discriminator that is trained on top of LM latent representations to control sentiment. fine-grained control of attributes is achieved by a simple gradient-based sampling mechanism

In[6], A fine-tuned model of BLEURT is used for pseudo-response selection. The dataset on which the experimentation is done is SGD and Weather benchmarks, and it is observed that the pro020 posed self-training approach improved the tree accuracy by 46%.

In [7], Global and local addressing scheme is used to structure the table content that uses a sequence to sequence technique for the model. Local addressing scheme is responsible for determining word in the table that would be used for the description generation while global addressing scheme determines word for summary generation. The LSTM model is used for generating natural language summarization of table. The dual attention mechanism present in the decoding phase is used for generating description from the table. Experimentation is done on the WIKIBO dataset .

In,[8],The paper showcases the use of a method the authors call GENPET which is based on pattern-exploiting training which also employs supervised learning . It was experimented on PEGASUS(Transformer encoder-decoder architecture) a spyware that yielded better results when compared to common finetuning across a set of different tasks and different training set sizes .GENPET only works for classification tasks and is observed to give good results in few shot settings. The paper explains GENPET to be a finetuning procedure created for generative language models that achieved goo data efficiency and used textual instructions and training examples.

In[9], an open-source toolkit is introduced that supports the broad set of text generation tasks transforms inputs into natural language. Texar extracts common patterns from the tasks and creates a library of reusable modules, and also allows arbitrary model architectures and algorithmic paradigms. The toolkit supports lot of large-scale pretrained models, TensorFlow and PyTorch. It is released under Apache License 2.0

In[10], The paper discusses how Pre-trained transformer models continue to increase in size and the different approaches to compress or utilize large pre-trained checkpoints into smaller and faster versions which retain the performance of the original models .Researches also show that subsets of trained teacher models can be extracted without affecting performance and therefore a "shrink and fine-tune" ("SFT") approach is discussed that extracts a student model from the maximally spaced layers of a fine-tuned teacher model . The hypothesis is that removing full layers will have minimal impact performance. Shrunken model is used to run the original fine-tuning procedure without any sort of modification .Experimentation is done on CNN and XSUM datasets.

In[11], The SLR a review paper that discusses 5 research aspects that associate with text generation. It reviews the quality metrics use for evaluating generated text, datasets used for training, languages on which the text generation is performed, and applications. Main aim of the survey is put together the relevant work in a systematic order and highlight important contributions from different researchers focusing on the past, present, and future trends. 90 primary studies are identified from 2015 to 2021 using the PRISMA framework.

In[12], Working of Bidirectional RNN is explored which makes use of two RNN layers and the sequence is looked in both forward and backward directions and output is combined .It is observed that BRNN can be trained without limiting it to using the input information. Experimentation uses the TIMIT database.

In[13],Different works on deep learning especially that associated with natural language processing are reviewed .Technologies like Recurrent Neural Networks (RNNs),Convolutional Neural Networks (CNNs),Variational Auto-Encoders (VAEs),Generative Adversarial Networks (GANs),Activation functions and Optimization techniques are explored and discussed .

In[14] a novel neural architecture is proposed which enables learning dependency without having to determine a fixed-length. It also doesn’t disrupt temporal coherence It observes a dependency 80% longer than that of RNNs and 450% longer for vanilla Transformers. Proposed model shows to achieve good performance on short and long sequences, and evaluates to be faster than other baseline systems.

In[15], The paper discusses error accumulation in Neural Machine Translation. context words were sampled from ground truth sequence and predicted sequence by the model during training. The approach achieved visible improvements in multiple datasets.

In[16], The paper introduces a new language representation model called BERT, (Bidirectional Encoder Representations from Transformers) designed to pretrain deep bidirectional representations from unlabeled texts. BERT is conceptually simple and empirically powerful.

In[17], a simple approach to implement conversational modeling task is explored .It uses sequence to sequence framework. The model works by predicting the next sentence given a previous set of sentences It extracts knowledge from both a domain specific dataset, and from a large, noisy, and general domain dataset of movie subtitles. the lack of consistency is a failure mode of the proposed model.

In[18]The paper observes that large gains on tasks involved in the understanding of natural language processing can be achieved by generative pre-training of a language model on a diverse corpus of unlabeled text, which should be followed by specific fine-tuning on each task. Task-aware input transformations are used to achieve effective transfer qand causing minimal changes to the model architecture.

In[19] The paper reviews different neural approaches to conversational AI that have come up in the last few years. Three categories of conversational systems are discussed that include question answering agents, task-oriented dialogue agents, and chatbots. Techniques like DNN-based response generation, neural Machine Reading Comprehension (MRC) model, Implicit ReasoNet (IRN) and d Neural Logic Programming (Neural LP)etc are discussed .

In[20], The paper proposes a generalized autoregressive pretraining method to overcome limitations of the language model known as BERT .The proposed technique called XLNet also enables learning context in both forward and backward directions.It has autoregressive formulation. It also uses ideas from Transformer-XL for the pretraining procedure .For some cases it is observed to outperform Bert.

In[21], A two parameter-reduction technique is proposed to over the issues of speed and memory consumption with BERT. The proposed model is said to scale better than the original model of BERT and uses self-supervised loss to focuses on modeling inter-sentence coherence which helps downstream tasks with multi-sentence inputs. Results of the model on GLUE, RACE, and SQuAD benchmarks are impressive

In[22], In this work, A new task: emotion-cause pair extraction (ECPE)is proposed to extract the potential pairs of emotions and the causes of the emotion in a file.It aims to overcome the shortcomings of traditional Emotion cause extraction (ECE)models .Here ,individual emotion extraction is performed and cause extraction is done using multi-task learningafter which emotion-cause. pairing and filtering is performed .

In [23], it makes use of the pcRU framework which is language generation framework to generate weather forecast . The models were evaluated on the basis of their output quality .Improved development time, increase in reusability of systems and better computational efficiency was observed over select NLG systems and happened to reduce computational expenses . Informed decisions were made during the generation phase.

In [24], A neural model is proposed to scale large and rich domains. Purpose of the system is to create large amounts of data specific to the input image given that uses OCR to recognize each character. A web crawler is used to gather data from datasets along the web mainly validated education websites in the text generation module. Main aim is to be able to provide information comparable to that found in text books for students of different educational field by taking in the syllabus as input

In [25],Automatic text generation from given input in the form of structured data is explored.Data2Text platform mainly describes the input text in the text generated and has three components namely model training, model revision and text generation. It uses semi-HMMs model to extract high quality templates and corresponding trigger conditions from parallel data

In [26]Proposed system is an entity-centric neural architecture for data-to-text generation where the model creates entity-specific representations which are dynamically updated. Text generated as output is based on the input that goes through entity memory representations which use hierarchical attention at each step. Output generated is coherent, concise and factually correct. The ROTOWIRE benchmark dataset is used for experimentation on the baseball domain .Results of the proposed model are observed to outperform competitive baselines in automatic and human evaluation.

In [27], The paper explores the idea of how decisions for word ordering and word choice in surface natural language generation can be automatically learned from annotated data . Four trainable systems are used in which NLG1 is kept as a baseline system for comparison and treated as a lookup table while NLG2 and NLG3 attempt to find the highest probability word sequence with respect to a maximum entropy probability model.NLG4 requires a dependency -style grammar for the fragments of phrases and is kept consistent with the rules of grammar.NLG4 implements dialogue strategy where order of words are dynamically modified The proposed system reorders words at runtime is basically represents practicality of learning the decisions for ordering the word and word choice. Thereafter we see a system that makes a conversation that has a more natural tone.

In [28] ,The paper explores the idea of using an ensemble network to achieve the goalof generating conclusion-supplement answers for non-factoid questions which is a difficult for the currently existing encoder-decoder frameworks.

The model proposed uses a neural network to store knowledge where it acknowledged that it is not possible to store all data present in the real world. The model uses sequence to sequence learning and end to end format is used for answers so as to get fluid responses . Context from conclusion decoder is used to create supplementary decoder states using attention mechanism and closeness of encoders output sequence is compared to separate outputs of conclusion and supplement decoders output sequence . Resulting answers match question . Experimentation done on datasets including “Love Advice” and “Arts & Humanities” categories observes more accurate results than other tested baseline models.

In [29], Proposed attention based model generates caption for photos given as input and uses pc vision strategies to grasp content of image and linguistic communication process for understanding image content into words. Experimentation is done on MS COCO , Flickr30k and Flickr8k dataset . Model uses a backpropagation technique to train and automatically fixes salient objects in generated output. Soft and hard attention mechanisms are used for generating image captions and BLUE and METOR metrics are used to evaluate state of the benchmark datasets.

In[30], a multi-turn open-domain chatbot trained end-to-end on data mined and filtered from public domain social media conversations is proposed . This neural network used has 2.6 billion parameters and minimizes perplexity of the next token. A human evaluation metric called Sensibleness and Specificity Average (SSA),is also proposed. It uses seq2seq model with Evolved Transformer (ET) and is trained on multi-turn conversations with the input .

4. ARCHITECTURE

4. ARCHITECTURE

4.1 PROJECT ARCHITECTURE

This project architecture shows the procedure for generating bulk text

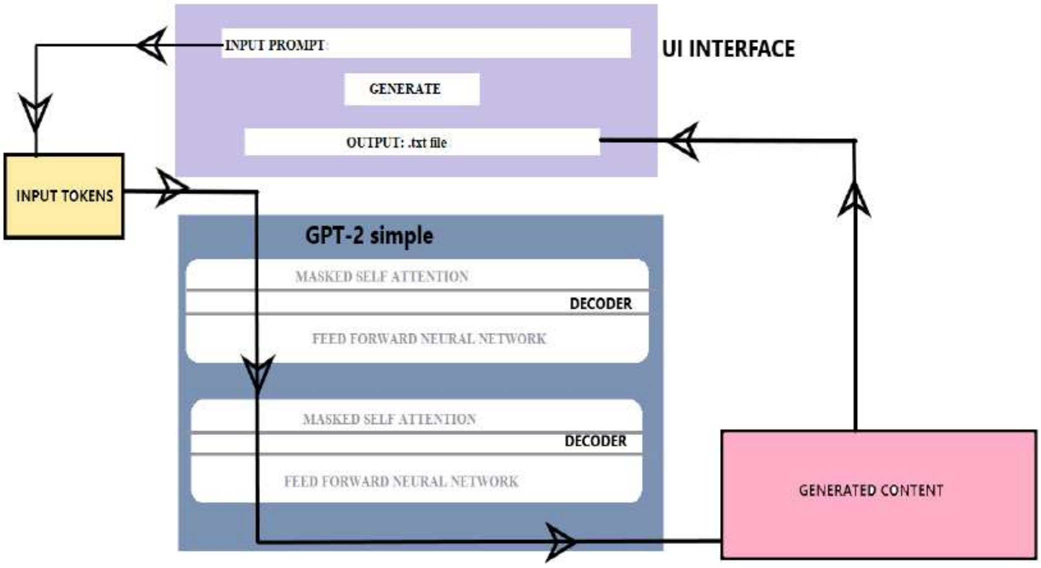


Figure 3.1: Project Architecture for Article writing

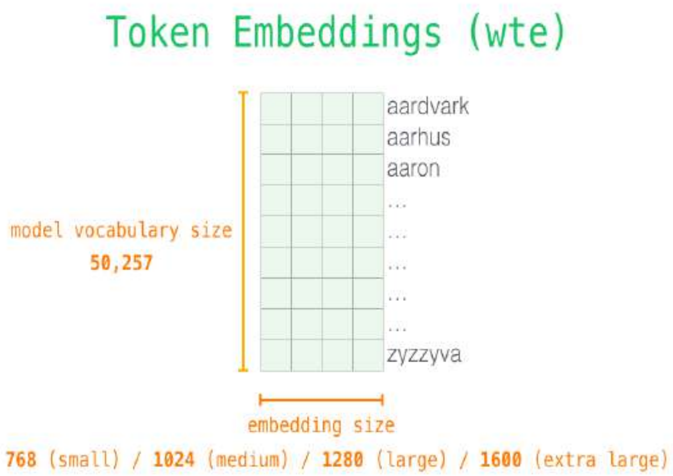
4.2 DESCRIPTION

- **Input prompt:** Input is taken from user in the form of text snippets or title.

Figure 4.2.1: Input text

- **Input tokens:**

  The given input text is converted into tokens then the model looks up the embedding of the input word in its embedding matrix – one of the components we get as part of a trained model.



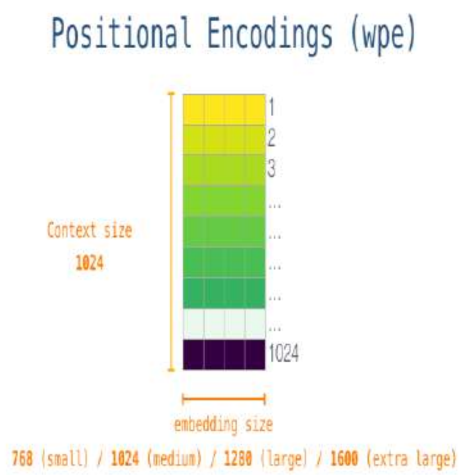


Figure 4.2.2: Token Embedding Matrix **Figure 4.2.3:** Positional Encoding Matrix

Before handing that to the first block in the model, we need to incorporate positional encoding – a signal that indicates the order of the words in the sequence to the transformer blocks.

Figure 4.2.4: Transformer block with decoder layers

- **Transformer Decoder Block:**

The first block can now process the token by first passing it through the self-attention process, then passing it through its neural network layer. the first transformer block processes the token, it sends its resulting vector up the stack to be processed by the next block. The process is identical in each block, but each block has its own weights in both self-attention and the neural network sublayers.

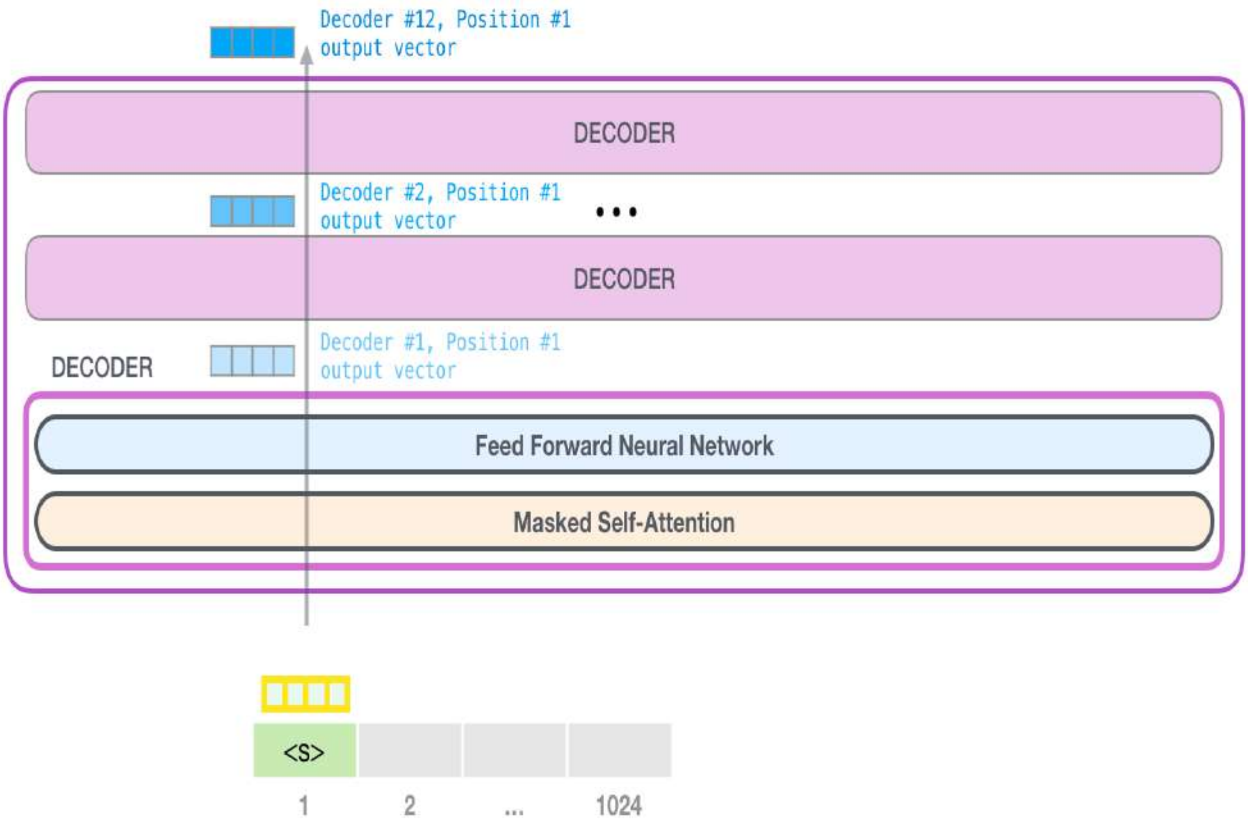


Figure 4.2.5: Internal structure of decoder layers

- **Masked Self-attention layer:**

It bakes in the model's understanding of relevant and associated words that explain the context of a certain word before processing that word (passing it through a neural network). It does that by assigning scores to how relevant each word in the segment is, and adding up their vector representation. It is processed along the path of each token in the segment. The significant components are three vectors:

Query: The query is a representation of the current word used to score against all the other words (using their keys). We only care about the query of the token we're currently processing.

Key: Key vectors are like labels for all the words in the segment. They're what we match against in our search for relevant words.

Value: Value vectors are actual word representations, once we've scored how relevant each word is, these are the values we add up to represent the current word.

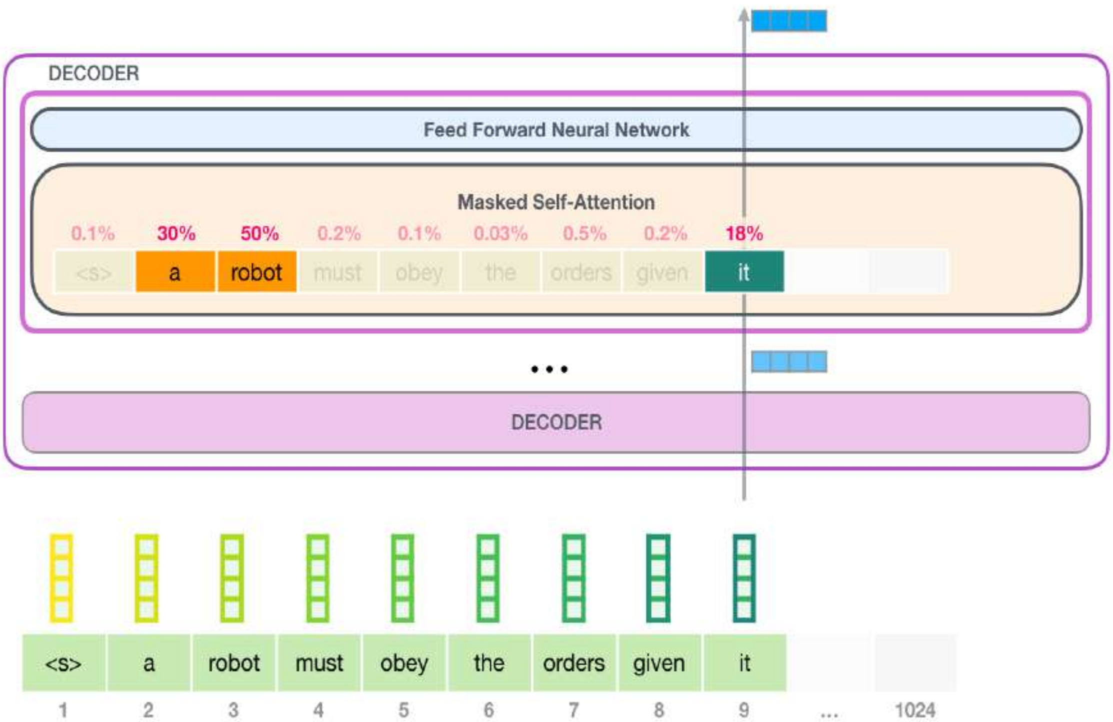


Figure 4.2.6: Internal structure of decoder layers

- **Feed forward neural network:**

Feedforward neural nets are complex network made up an input layer that accepts information, hidden layers that capture the hidden correlations between each data point, and an output layer which transmit information.

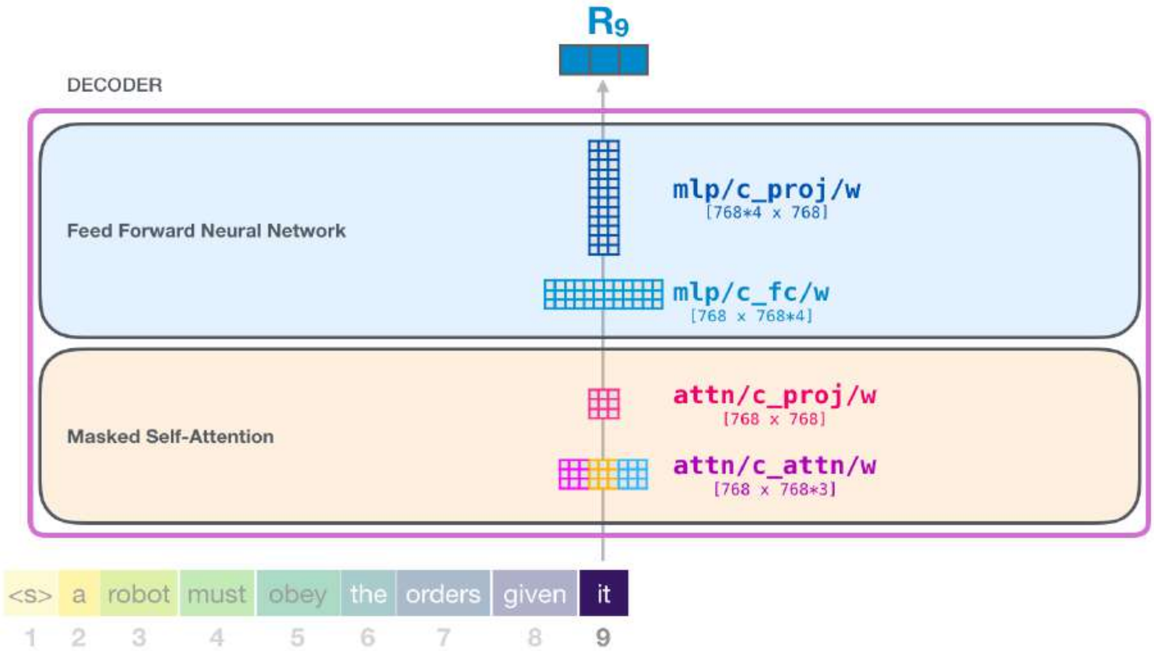


Figure 4.2.7: Internal structure of decoder layers

- **Output:** Generated content can be downloaded in a .txt file.

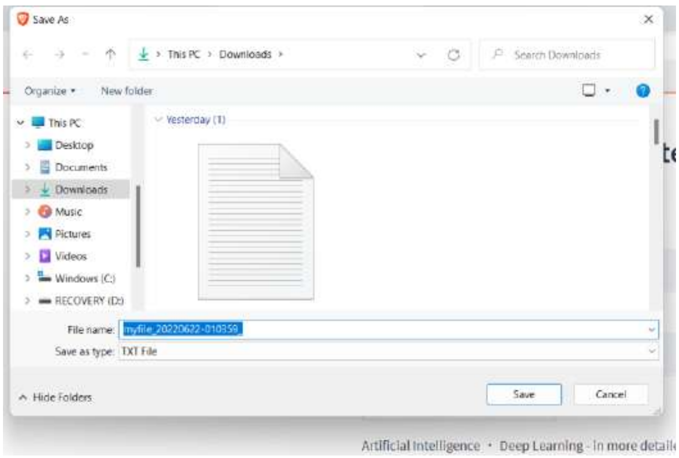


Figure 4.2.8: Downloade Output

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically three actors who are the user and admin.User will give the title of the article or an input snippet for generating an article. Admin makes updates in the model by fine tuning it.

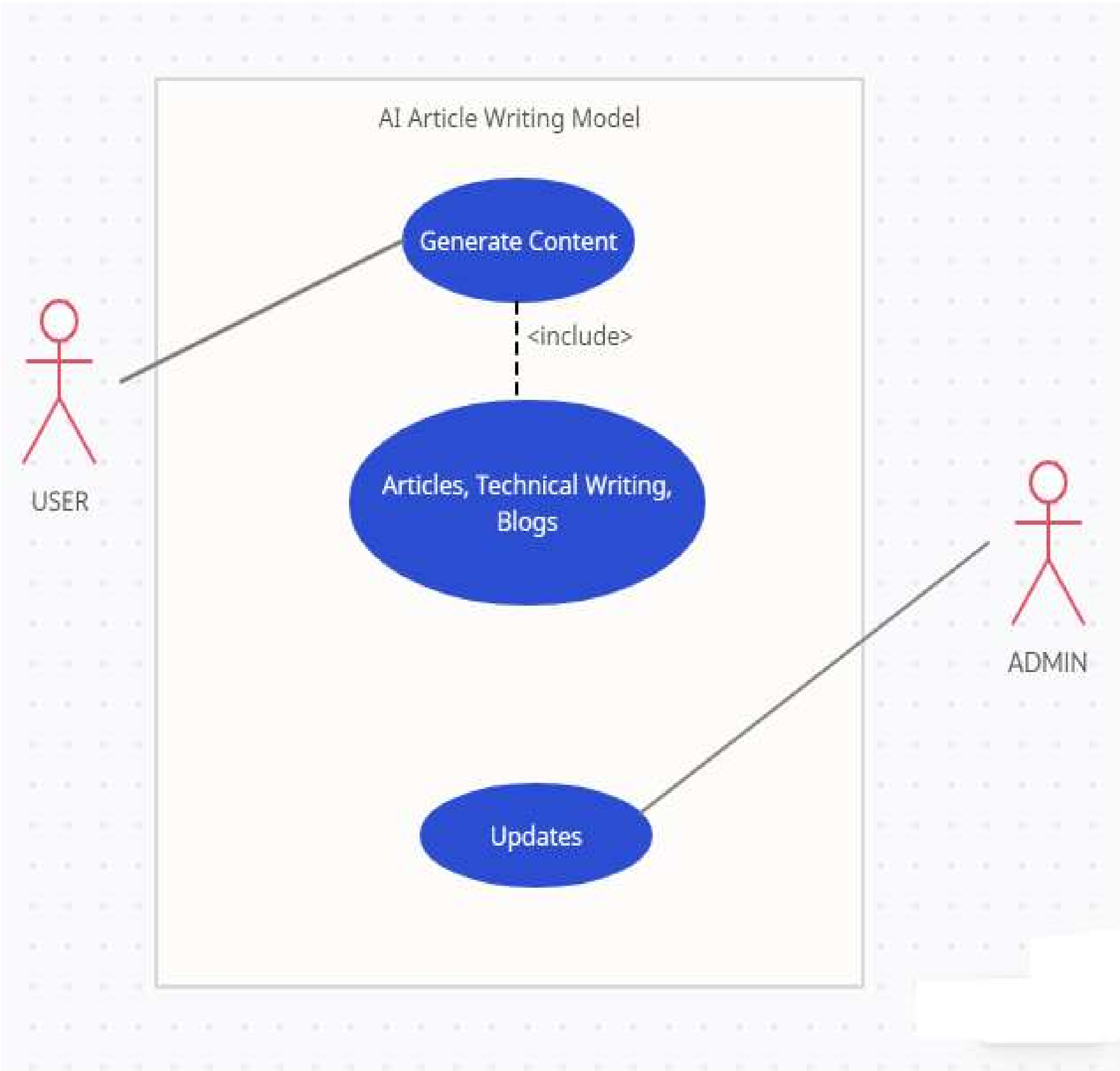


Figure 4.3: Use Case Diagram for Article writing

3.4 SEQUENCE DIAGRAM

A **sequence diagram** or system **sequence diagram** (SSD) shows object interactions arranged in time sequence in the field of software engineering. Here input is given through an interface and is processed by the proposed model which then generates related and specific to topic content. The generated content can be downloaded by the user through the interface and is available in a .txt file. The sequential process is represented below.

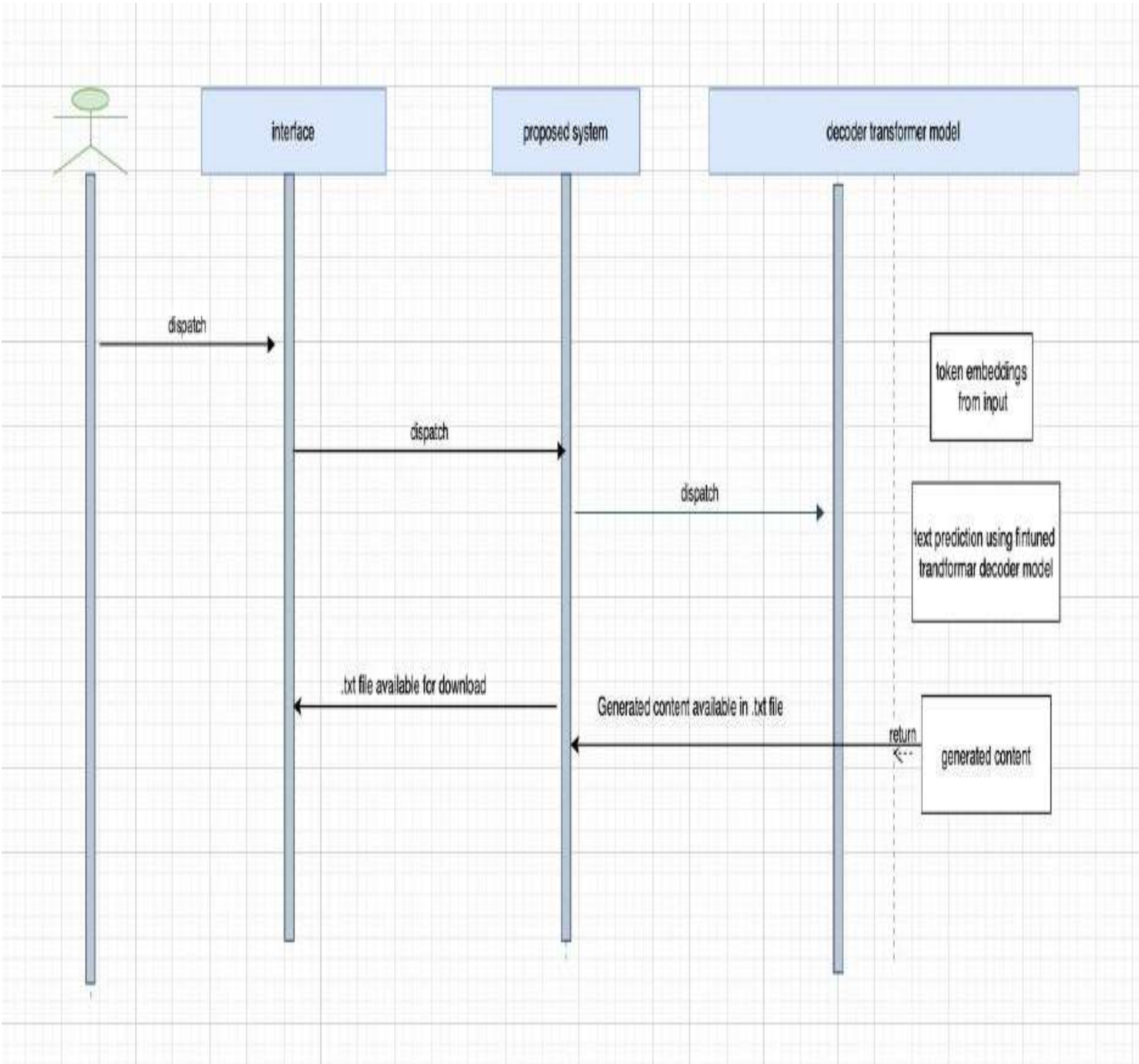


Figure 4.4: Sequence Diagram for Article writing

3.5 ACTIVITY DIAGRAM

It describes the flow of activity states. In our Article writer the first activity is entering article title or text then our model read the input text and generates the specific text related to the input text . Input specific text is generated in .txt file format.

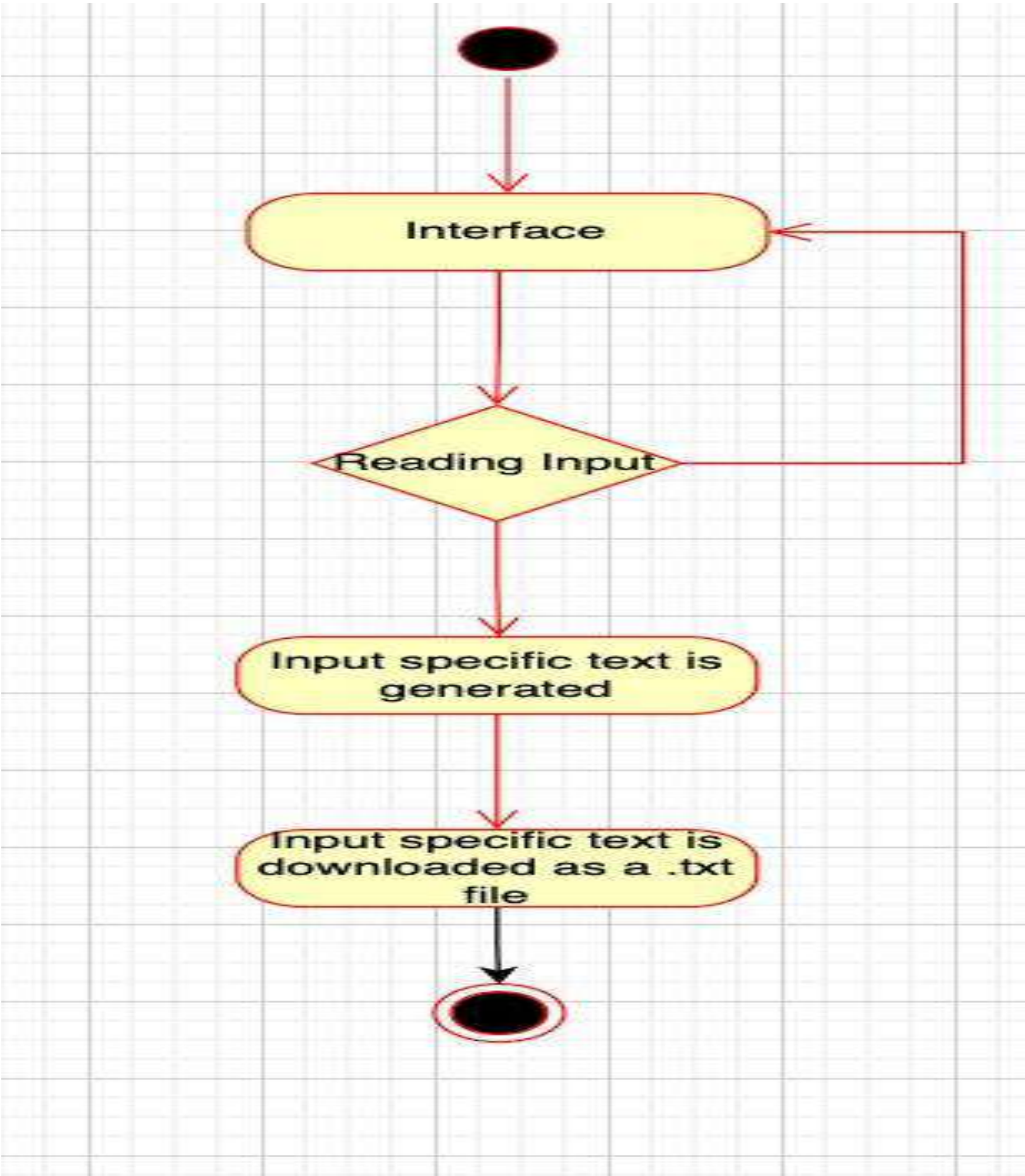


Figure 4.5: Activity Diagram for Article writing

5. IMPLEMENTATION

5. IMPLEMENTATION

5.1 SAMPLE CODE

```
import gpt_2_simple as gpt2
from datetime import datetime
from google. colab import files
import tensor flow as tf
from google.colab import drive drive.mount('/content/drive')
gpt2.download_gpt2 (model_name="124M")
gpt2.mount_gdrive()
file_name= "paridhi_train.txt"
gpt2.copy_file_from_gdrive(file_name)
tf.compat.v1.reset_default_graph()
    sess = gpt2.start_tf_sess()
    gpt2.finetune(sess,
                  dataset file_name,
                  model_name='124M',
                  steps=4000,
                  restore_from='latest',
                  run_name='run1',
                  print_every-1000,
                  learning_rate=1e-4,
                  sample_every=1000,
                  overwrite=True,
gpt2.copy_checkpoint_to_gdrive (run_name='run1')
gpt2.copy_checkpoint_from_gdrive (run_name='run1')
tf.compat.v1.reset_default_graph()
sess = gpt2.start_tf_sess()
gpt2.load_gpt2(sess, run_name='run1')
gpt2.generate(sess,
              length=500,
```

```
                    temperature=0.7,
                    top_k=40,
                    top_p=0.9,

                    truncate='<|endoftext|>',
                    prefix="Artificial Intelligence for automobiles",
                    nsamples=1,
                    batch_size=1
gen_file =  'gpt2_gentext_{: %Y%m%d %HXMXS}.txt'.format(datetime.utcnow())

   gpt2.generate_to_file (sess,
                          destination_path=gen_file,
                          length=500,
                          temperature-0.7,
                          nsamples=1,
                          batch_size=1
files.download (gen_file)
```

5.2 UI Code

```
import os
import warnings
import re
import numpy as np
import pandas as pd
import seaborn as sns
from keras_gpt_2 import load_trained_model_from_checkpoint, get_bpe_from_files,
generate
import gpt_2_simple as gpt2
import streamlit as st
import warnings
import base64
import time

import streamlit.components as stc
timestr = time.strftime("%Y%m%d-%H%M%S")

class FileDownloader(object):

    def __init__(self, data, filename='myfile', file_ext='txt'):
        super(FileDownloader, self).__init__()
        self.data = data
        self.filename = filename
        self.file_ext = file_ext

    def download(self):
        b64 = base64.b64encode(self.data.encode()).decode()
        new_filename = "{}_{}_.{}".format(
            self.filename, timestr, self.file_ext)
        st.markdown("#### Download File ###")
```

```
        href = f'<a href="data:file/{self.file_ext};base64,{b64}"
download="{new_filename}">Click Here!!</a>'
        st.markdown(href, unsafe_allow_html=True)

warnings.filterwarnings("ignore")
model_folder = 'G:/work gpt2/checkpoint/run1'
config_path = os.path.join(model_folder, 'hparams.json')
checkpoint_path = "G:/work gpt2/checkpoint/run1.ckpt"
encoder_path = os.path.join(model_folder, 'encoder.json')
vocab_path = os.path.join(model_folder, 'vocab.bpe')

st.header("High Performance Content Writer Using Deep Learning")
print('Load model from checkpoint...')
model = load_trained_model_from_checkpoint(config_path, checkpoint_path)
print('Load BPE from files...')
bpe = get_bpe_from_files(encoder_path, vocab_path)
print('Generate text...')
text = st.text_input("Enter the Text")
length = st.number_input("Enter the Length", step=1)

if st.button('Click to run the content writer'):
    output = generate(
        model, bpe, [text], length, top_k=40, temperature=0.7)
    st.write(output[0])
    download = FileDownloader(output[0]).download()
```

6. TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNITTESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent.

Integration testing is specifically aimed at exposing the problems that arise from the combination of component.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested area valuable as specified by the business and technical requirements, system documentation, and user manuals.

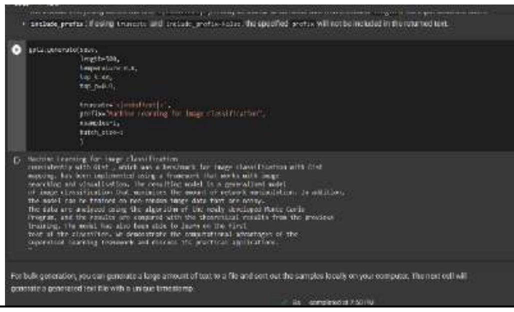
Functional testing is centered on the following items:

Valid Input : Identified classes of valid input must be accepted.
Input : Identified classes of invalid input must be rejected
Functions : Identified functions must be exercised.
Output : Identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes.

6.3. TEST CASES

Table 6.3.1 Testing

Test case ID	Test case name	Purpose	Test Case	Output
1	Content generation test1	To check if the model gives the content related to the title	User enter the title of the article to be generated	Model generates content related to the title 
2.	Content generation test2	To check whether the model generates the specified length in accordance	User specifies the length of the article to be generated	Model generates article according to specified length 
3.	Content generation test3	To check if the model produces content that is not repetitive	User can read the content generated by the model	The content generates by the model is not ambiguous
4	Content generation test4	To check if the provided information is factual and not gibberish	User can read the content generated by the model	The content produced by the model is trained on a 70mb factual data

7. RESULTS

6. RESULTS

7.1 HOME PAGE

Screenshot 7.1: User Interface

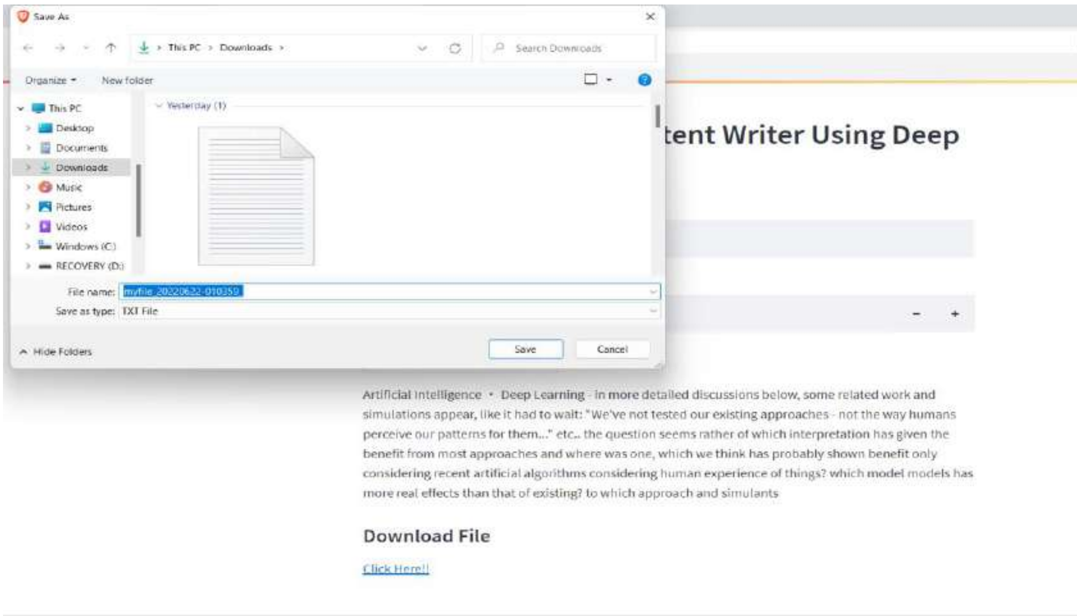
7.2 INPUT TEXT

Screenshot 7.2: User Enters the title of the content to be generate

7.3 Text Generation

Screenshot 7.3: Result of High Performance Content Writer

7.4 DONWLOAD THE GENERATED CONTENT



Screenshot 7.4: Generated content can be downloaded by the user through the interface and is available in a .txt file

8. CONCLUSION

8. CONCLUSION & FUTURE SCOPE

8.1 PROJECT CONCLUSION

Applications of artificial intelligence to generate content is the next big thing that will create its own place in content creation. Content creation still represents a standing challenge for deep-learning NLP. Even more so this task is applied to a domain-specific corpus that are different from the pre-training. highly technical, or contains low amount of training materials. Nevertheless, we have here illustrated that the text-to-text, multi-loss training strategy could be used to fine-tune a pre-trained language model such as GPT-2 for content generation The fact that the GPT2 generated content showing good readability and succinct information coverage are not reflected. The result is interpretable and reasonable, even though it is not near human-level performance. We think that our model could benefit from further training. This should make the model more accurate its ability. The use of artificial intelligence in content generation is not a new development, but it is becoming more popular as the technology becomes more sophisticated. There are a number of different ways that artificial intelligence can be used to generate content. One way is to use artificial intelligence to analyze a body of text and then to generate new text based on the analysis. This is sometimes called “natural language generation”. This approach can be used to create summaries of documents or to generate new content based on a set of data. Another way that artificial intelligence can be used to generate content is to use it to create images or videos. This approach is often called “computer vision”. This approach can be used to create images for stock photo services or to create videos for advertising. As AI grows more sophisticated, figuring out how to enable the good uses without the bad ones will be one of our biggest challenge. At the end of the day, AI should be used as a tool to improve and accompany the content writing process, not be the sole source of copy.

8.2 FUTURE SCOPE

The future of article writing is not about whether there will be humans or AI writers, but how these two will work together. Content generation has gained wide popularity because a profusion of applications uses them, and there is abundant availability of text online thanks to social media, news outlets, and other sources with enormous usage of text. A few applications which are benefited from text generation include generating and predicting character/word/sentence while typing an email or chatting, chatbot, movie/drama scriptwriting, poetry generation, and many other applications. Moreover, text generation has also been attracting the attention of researchers in the application area of education, industry, and social networks to provide an insight view on different aspects of the approaches. . Content writers should use AI writing assistants to generate content ideas at scale and focus on what they are best at - creativity and emotions. The future of AI is bright, and it will continue to evolve alongside humans. The balance between human and AI will be maintained as well, with humans still in control of the process, but with the help of AI assistants.

9. REFERENCES

9. BIBLIOGRAPHY

9.1 REFERENCES

[1] Y. Qu, P. Liu, W. Song, L. Liu and M. Cheng, "A Text Generation and Prediction System: Pre-training on New Corpora Using BERT and GPT-2," *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2020, pp. 323-326, doi: 10.1109/ICEIEC49280.2020.9152352.

[2] Kieuvongngam, Virapat, Bowen Tan and Yiming Niu. “Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2.” *ArXiv* abs/2006.01997 (2020): n. page.

[3] Bartoli, Alberto and Eric Medvet. “Exploring the Potential of GPT-2 for.” (2020).

[4] Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever. “Language Models are Unsupervised Multitask Learners.” (2019).

[5] Dathathri, Sumanth and Madotto, Andrea and Lan, Janice and Hung, Jane and Frank, Eric and Molino, Piero and Yosinski, Jason and Liu, Rosanne(2019) “Plug and Play Language Models: A Simple Approach to Controlled Text Generation, accessed 16,may,2022, https://arxiv.org/abs/1912.02164

[6] Mehta, Sanket Vaibhav and Rao, Jinfeng and Tay, Yi and Kale, Mihir and Parikh, Ankur P. and Strubell, Emma(2021) “Improving Compositional Generalization with Self-Training for Data-to-Text Generation”, accessed 25,June,2022, https://arxiv.org/abs/2110.08467

[7]. Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, Zhifang Sui “Table- to- Text Generation by Structure- Aware Seq2seq Learning” The Thirty-second AAAI Conference on Artificial Intelligence (AAAI-18) 2018

[8] Schick, Timo and Schütze, Hinrich(2020) “Few-Shot Text Generation with Pattern-Exploiting Training”, https://arxiv.org/abs/2012.11926

[9] Hu, Zhiting and Shi, Haoran and Tan, Bowen and Wang, Wentao and Yang, Zichao and Zhao, Tiancheng and He, Junxian and Qin, Lianhui and Wang, Di and Ma, Xuezhe and Liu, Zhengzhong and Liang, Xiaodan and Zhu, Wangrong and Sachan, Devendra Singh and Xing, Eric P.(2018) “Texar: A Modularized, Versatile, and Extensible Toolkit for Text

Generation”, arXiv:1809.00794v2

[10] Shleifer, Sam and Rush, Alexander M.(2020) “Pre-trained Summarization Distillation”, https://arxiv.org/abs/2010.13002v2

[11] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota and A. Soomro, "A Systematic Literature Review on Text Generation Using Deep Neural Network Models," in *IEEE Access*, vol. 10, pp. 53490-53503, 2022, doi: 10.1109/ACCESS.2022.3174108.

[12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[13] Touseef Iqbal, Shaima Qureshi(2022)“The survey: Text generation models in deep learning”, Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 6, Part A, Pages 2515-2528,ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2020.04.001.(https://www.sciencedirect.com/science/article/pii/S1319157820303360)

[14] Dai, Zihang and Yang, Zhilin and Yang, Yiming and Carbonell, Jaime and Le, Quoc V. and Salakhutdinov, Ruslan(2019) “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context”, https://arxiv.org/abs/1901.02860

[15] Zhang, Wen and Feng, Yang and Meng, Fandong and You, Di and Liu, Qun(2019) “Bridging the Gap between Training and Inference for Neural Machine Translation”, https://arxiv.org/abs/1906.02448v2

[16] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina(2018) “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, https://arxiv.org/abs/1810.04805v2

[17] Vinyals, Oriol and Le, Quoc(2015) “A Neural Conversational Model”, https://arxiv.org/abs/1506.05869v3

[18] Radford, Alec and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training.” (2018).

[19] Gao, Jianfeng and Galley, Michel and Li, Lihong(2018) “Neural Approaches to Conversational AI”, https://arxiv.org/abs/1809.08267v3

[20] Zhilin Yang and Zihang Dai and Yiming Yang and Jaime Carbonell and Ruslan Salakhutdinov and Quoc V. Le(2019), “XLNet: Generalized Autoregressive Pretraining for Language Understanding”, https://doi.org/10.48550/arXiv.1906.0823

[21] Lan, Zhenzhong and Chen, Mingda and Goodman, Sebastian and Gimpel, Kevin and Sharma, Piyush and Soricut, Radu(2019) “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”, https://arxiv.org/abs/1909.11942v6

[22]. Xia, Rui and Ding, Zixiang(2019) “Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts”, https://arxiv.org/abs/1906.01267v1

[23]. Anja Belz,”Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-Space Models”, Natural Language Engineering 1 (1): 1-26. Printed in the United Kindom

[24]. Lebert,R; Grangier,D; and Auli,M. 2016.,”Neural text generation from structured data with application to the biography domain”, arXiv preprint arXiv:1603.07771

[25]. Longxu Dou , Guanghui Qin , Jinpeng Wang , Jin-Ge Yao3 , and Chin-Yew Lin3,” Data2Text Studio: Automated Text Generation from Structured Data”, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), pages 13–18

[26]. Ratish Puduppully and Li Dong and Mirella Lapata “Data-to-text Generation with Entity Modeling”, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages.

[27]. Adwait Ratnaparkhi,”Trainable approaches to surface natural language generation and their application to conversational dialog systems”, 2002 Computer Speech & Language, 16(3): 435-455.

[28]. J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li, “Neural generative question answering,” in Proc. IJCAI,2016, pp. 2972–2978.

[29]. Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, arXiv: 1502.03044v3 [cs.LG] 19 Apr 2022.

[30] Adiwardana, Daniel and Luong, Minh-Thang and So, David R. and Hall, Jamie and Fiedel, Noah and Thoppilan, Romal and Yang, Zi and Kulshreshtha, Apoorv and Nemade, Gaurav and Lu, Yifeng and Le, Quoc V.(2020) “Towards a Human-like Open-Domain Chatbot”, https://doi.org/10.48550/arXiv.2001.09977

9.2 WEBSITES

https://minimaxir.com/2019/09/howto-gpt2/

https://en.wikipedia.org/wiki/GPT-2

https://colab.research.google.com/drive/1Y_KrkEhNlTh6fmg1wb VcL0FmCWsj643S

https://www.google.com/search?q=transformer-+decoder+explanied&oq=transformer-+decoder+explanied&aqs=chrome..69i57.833876j0j1&sourceid=chrome&ie=UTF-8

https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0

https://jalammar.github.io/illustrated-gpt2/

https://towardsdatascience.com/how-to-fine-tune-gpt-2-so-you-can-generate-long-form-creative-writing-7a5ae1314a61

https://medium.com/swlh/everything-gpt-2-2-architecture-comprehensive-57129fac417a

https://github.com/abishawinslet1101

https://github.com/530p

https://github.com/Suchith1999