Jadavpur University

Poem Generation in Bengali

By

Srabana Ghoshal

Roll No.:002010502001

Registration No. -154125 of 2020-2021

Session: 2020-2022

This dissertation is submitted for the degree

of Master of Engineering

Under the Guidance and Supervision of

Dr. Sudip Kumar Naskar

Department of Computer Science and Engineering

Jadavpur University

Kolkata-700032

June 2022

Declaration of Authorship

I, hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of the Master in computer science and Engineering studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by thesis rules and conduct, I have fully cited and referenced all materials that are not original to this work.

Signature:_____ Date:_____

Name: Srabana Ghoshal

Examination Roll No.:M4CSE22001

Class Roll No. : 002010502001

Registration number: 154125 of 2020-2021

Session: of 2020-2022

Thesis Title: Poem Generation in Bengali

Certificate of Recommendation

This is to certify that the dissertation entitled “Poem Generation in Bengali” has been carried out by Srabana Ghoshal (University Registration No: 154125 of 2020-21, Roll No :002010502001 under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master in Computer Science and Engineering. The research results presented in the thesis have not been included in any other thesis submitted for the award of any degree in any other university or Institute.

Dr. Sudip Kumar Naskar
Thesis Supervisor
Dept. Of Computer Science and Engineering
Jadavpur University

Signature: _____

Prof. Anupam Sinha
Head, Dept. Of Computer Science and Engineering
Jadavpur University

Signature: _____

Prof. Chandan Majumdar
Dean, Faculty of Engineering and Technology
Jadavpur University

Signature: _____

Certificate of approval

This is to certify that the thesis entitled “Poem Generation in Bengali” is a bonafide record of work carried out by Srabana Ghoshal (University Registration No: 154125 of 2020-2021, Roll No:002010502001) in the partial fulfillment of the requirements for the award of the degree of Master in Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University, During the period of Sept 2021 to June 2022. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, the opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

External Examiner:

Signature :_____ Date:_____

Dr. Sudip Kumar Naskar

Dept. Of Computer Science and Engineering

Jadavpur University,Kolkata-700032

Signature :_____ Date:_____

Acknowledgement

I would like to express my sincere gratitude to my advisor, Dr. Sudip Kumar Naskar , for his continuous motivation, guidance, and patience throughout my thesis work. I have been very lucky to have an advisor who cared so much about my work. More importantly, the scientific and personal support along with his valuable suggestions softened the journey .I am also thankful to all other NLP lab members for their continuous support.

Srabana Ghoshal

Signature :_____

Abstract

Natural language generation is a subfield of artificial intelligence (AI), a software process that converts data into plain text content. Technology can tell a story ,poems - just like a human analyst- by writing sentences and paragraphs,or making comments.

NLG is one of the fastest growing technologies being adopted in the enterprise. There are many use cases for NLG, but where it is seen to be most effective it is deployed to automate time - intensive data analysis and reporting activities. Although a number of approaches have been introduced in the past (e.g. story grammar, story schema and autonomous agents), they all rely heavily on handwriting resources,.The traditional models are are found to be fluent with short texts.But when it comes to practicality long texts are also necessary,which places great limits on its scalability and usability.So, new models have been proposed based on the traditional technologies as backbone to find improved performance in terms of length,coherence as well as relevance .

Table of Contents

Chapter 1

Introduction

1.1 Overview

The goal of text generation is to make machines express themselves in human language. It is one of the most important yet challenging tasks in natural language processing (NLP). Since 2014, various neural encoder-decoder models pioneered by Seq2Seq have been proposed to achieve the goal by learning to map input text to output text. However, the input text alone often provides limited knowledge to generate the desired output, so the performance of text generation is still far from satisfactory in many real-world scenarios. The survey of NLG is timely in view of the changes that the field has undergone over the past two decades, especially in relation to new (usually data-driven) methods, as well as new applications of NLG technology. This survey therefore aims to (a) give an up-to-date synthesis of research on the core tasks in NLG and the architectures adopted in which such tasks are organized;(b) highlight a number of recent research topics that have arisen partly as a result of growing synergies between NLG and other areas of artificial intelligence; (c) draw attention to the challenges in NLG evaluation, relating them to similar challenges faced in other areas of NLP, with an emphasis on different evaluation methodsand the relationships between them. People use stories to communicate effectively with one another. As humans, we engage with well-told stories and comprehend more information from stories(Suzuki et al., 2018). However, when it comes to automatic natural language generation still have a long way to go.

The importance of computational narrative is that it can improve human interaction with intelligent systems. Computational narrative research involves literature text understanding, literature text representation, and literature text generation. Many surveys were written on different facets of computational storytelling. (Gervás, 2009) provides a chronological summary of storytelling systems focusing on computational creativity, measured using metrics including the stories' novelty and the users' involvement in the storytelling process. (Riedland Bulitko, 2013) focuses on interactive intelligence, a digital interactive storytelling experience where users interact with

the computational system to build storylines.(Riedl, 2016) discusses human-centered computational narrative and how it can improve artificial intelligence applications.Interest in the

development of automatic methods for generating poetry dates back to the 1960s, even before computers were accessible to everyone. A commonly cited example is the creation of a large number of new poems by interchanging the lines in a set of poems, always respecting the relative position of each line within a stanza (Queneau, 1961). Early approaches to experimental poetry applied a set of combinatory processes to existing poems, in order to generate new ones. This challenge was embraced by different groups, such as the Portuguese movement of Experimental Poetry (PO.EX, e.g. the collection by Torres and Baldwin (2014)), or the French Atelier of Literature Assisted by Math and Computers (ALAMO, see Oulipo (1981)). A notable work of the latter includes the rim baudelaires, where the structure of a poem by Rimbaud is filled with vocabulary from Baudelaire's poems. At the time, interested people were mostly poets or researchers from the domain of Humanities.

Mainly researchers in the domain of Artificial Intelligence (AI) and, specifically,Computational Creativity (Colton and Wiggins, 2012) since the works of Gervas (2000) and ´ Manurung (2003), the development of "intelligent" poetry generation systems has seen a significant increase. These systems are not limited to rewriting text in a poetic form. They are often knowledge intensive natural language generation systems that deal with several levels of language (e.g. phonetics, lexical choice, syntax and semantics) to produce aesthetically-pleasing text with creative value.

What makes this task more interesting is that some levels of language do not have to be strictly addressed. Writing poetic text does not have to be an extremely precise task (Gervas, 2000), as several rules, typically present in the production of natural language, need to be broken (Manurung, 2003). On the other hand, poetry involves a high occurrence of interdependent linguistic phenomena where features such as metre, rhyme, and the presence of figurative Language plays an important role. For instance, it might be ok to transmit a less clear message, in a trade-off for a pleasant sound given by a highly regular metre. In fact, the latter is easier to achieve by a computer, while transmitting a clear message can be highly challenging, especially when constrained by the previous features.Besides the involved challenge, poetry generators can be useful for applications in different areas, such as education or electronic entertainment.

1.2 Bengali Poetry

Bengali (also known as Bangla) is the world's seventh most widely spoken language. It is related to Sanskrit and is part of the Indo-Aryan language family. Bengali is India's second most widely spoken language and Bangladesh's official language. Bengali poetry has a long history dating back to the 10th century, and current Bengali poetry is based on Sanskrit. As the first non-European Nobel Laureate in Literature, Rabindranath Tagore (1861–1941) was the pioneer who founded the firm basis of modern Bengali poetry and is best recognised for his poems.

Bengali is partially phonemic, as are all Modern Indo-Aryan languages descended from Sanskrit. That is, its pronunciation style is influenced not just by orthographic information, but also by POS and semantic information.

Languages that are partially phonemic use writing systems that are halfway between strictly phonemic and non-phonemic. Sanskrit orthography is still used in Bengali and many other modern Indo-Aryan languages, albeit the sounds and pronunciation norms have changed to varied degrees.The modern Bengali script contains the characters (known as akṣara) for seven vowels (/i/ /u/, /e/, /o/,/æ/, /O/, /a/), four semi-vowels, (/j/, /w/, /e̯/, /o̯/), and thirty consonants.

A phonetic group of Bengali consonants is called a borgo (বর্গ).

In Sanskrit poetry, there are three sorts of rhymes, depending on whether the rhyme is on the first syllable of each line (adiprsa), the second syllable (dviteeyakshara prsa), or the final syllable of the line (antyaprsa). Antyaprsa, also known as tail-rhyme or end-alliteration in English and anto-mil in Bengali poetry, is the most crucial rhyme for our needs.

Chapter 2

Literature survey

2.1 Neural Model based NLG

Automatic Generation of News Comments Based on Gated Attention Neural Networks [4] presents a new,challenging task in NLG. This task is helpful to comprehend human languages, like how people write news comments, what people pay attention to, and how people express it. Moreover, it is also a useful exploration for the future task of automatic generation of comment articles. Additionally, these generated comments are beneficial for companies. For example,a comment writing assistant which generates some candidate comments for users could be built. Users could select one and refine it, which makes the procedure more user-friendly.

There are two challenges mainly in comment generation.Firstly, contextual relevance between comments and news should be ensured.Complex contextual information should be dealt with ,as a news event contains a lot of aspects like time,location,event etc.The generated comment should be relevant to these aspects of news semantics. Secondly, there is a need to generate diversified comments for every news. As, different users comment on different aspects of the same news article.These are really challenging.

The task of generating comments is achieved by GANN and attention mechanisms. The gated attention mechanism is used to get selective and self adaptive news context, which in turn is more focused to the contextual relevance. Diverse comments from multiple topics and different aspects from random samples are generated to find the relevance control. Generative adversarial nets are then used to improve the comment generator in order to generate more natural comments. Experiments show that on a real dataset to generate comment GANN performs better than existing comment generation methods.

The task of generating comments is achieved by GANN and attention Mechanisms. The gated attention mechanism is used to get selective and self adaptive news context, which in turn is more focused to the contextual relevance. Diverse comments from multiple topics and different aspects

from random samples are generated to find the relevance control. Generative adversarial nets are then used to improve the comment generator in order to generate more natural comments. Experiments show that on a real dataset to generate comment GANN performs better than existing comment generation methods.

The task of automatic generation of news comments using the Gated Attention Neural Network model(GANN)is studied . GANN is built on the Encoder-Decoder framework .There are three mechanisms applied to improve the performance.The gated attention mechanism improves the performance significantly by enhancing the contextual relevance.The relevance control generates diverse comments and the discriminator boosts the GANN by the adversarial training.Experiments show that GANN generated news comments are close to human generated comments.

A Creative Computing based Inspiration Assistant to PoemGeneration [5]. A Creative Computing based Inspiration Assistant to Poem Generation Creativity is an ability of human beings.Enhancing an individual's creativity has a lot to do with the individual's mind. The brain functionalities ,cognitive ability, connections between parts of the brain and information exchange between the left and right part of the brain improves .This in turn generates better ideas helping in the overall development of not only the individual , but also the society.Instilling creativity in artificial entities using artificial intelligence or machine learning is a challenging task.Prediction of result or to emulate human brain is so far achieved but not creativity in machines.Machines are not able to achieve real creation till now.So, to focus on simulation of The human brain to make machines creative is still a theme in the creativity domain. But, knowledge combination can contribute and help the inspirational process.The innate motivation and external simulation for creativity from psychology can be abstracted.This can be used as a word linking process using Semantic Web technology to make software understand the context and thus generate output in accordance to that.

A feedback loop can further be constructed to make the software adaptive.The poem should be analyzed based on features of the poem database.There are some fixed features and some flexible features. The Haiku poem can be chosen such that the flexible features can be changed and extracted from poem structure.The poem framework may include rhyme and specific vocabulary or nouns that are unchangeable. On the basis of features of creativity, the transformed poem should be surprising and useful.The words used for expressing characteristics should fit well.

The generated output is a framework of a poem with changed words relevant to the context and meaning of the input poem. In the process of word linking,the user could input words according to preference so that the output poem contains specific words .This word from the user acts as an external simulation function .This can be used to find a list of positive and negative relevant words that could be stored in an associated word dataset for future use.

The original word and the related words are delivered to the Semantic Web function to discover the knowledge in words.This provides intrinsic motivation.The user can further choose a relevant word to connect with the original word and can choose a word that has an opposite meaning to generate the poem as well.The last step is to focus on user satisfaction about the generated Haiku Poem. The satisfaction degree can be divided into five levels .

The original word and the related words are delivered to the Semantic Web function to discover the knowledge in words.This provides intrinsic motivation.The user can further choose a relevant word to connect with the original word and can choose a word that has an opposite meaning to generate the poem as well.The last step is to focus on user satisfaction about the generated Haiku poem.The satisfaction degree can be divided into five levels .If the satisfactory degree is more than 60% then the poem is considered useful and novel,which can provide the user with some inspiration. Else the poem is sent back to the word linking process to modify features and generate another poem.This recursive process finally leads to finding a poem the user finds useful in terms of inspiration and usefulness.The approach designed in this paper, includes database establishment layer, feature analysis layer, replacement layer and feedback loop layer. This can achieve the goal to inspire people in writing poems. The percentage of how users are satisfied about results will not only be used to judge the quality of current results but also be utilized in the process of changing a poem.The approach aims to change the general poems in different ways which depend on the users satisfaction about the current results and this will make the results in accordance with users' requirement in a more accurate way.

The Creative Turn: New Challenges for Computing [6].Poem Generation using Transformers and Doc2Vec Embeddings The paper presented a method of generating poems from given input poem seed using transformers and doc2vec embeddings.

Three transformers were trained for 1500 steps, 1900 steps, and 100 steps, respectively, using a dataset composed of a single verse from the long poem dataset, the short poem dataset, and the long poem dataset (single verse). The models were allowed to generate 10 samples of up to 50tokens each. The Samples were split if a delimiter token appears as a substring.

The corresponding doc2vec model was also trained using each of the mentioned datasets to evaluate the output poems. The doc2vec models for both the long poem and short poem were trained for 1000 epochs, while the single poem was only trained for 400 epochs.

The generation set-up produced truncated outputs. Using three input rhyme seeds, each poet transformer was allowed to produce multiple outputs, these were checked between documents in the doc2vec model with each output having the highest cosine similarity. The results were matched and the hit rate was calculated to obtain outputs that could be identified as belonging to their respective poet. The method benefits from a pre-trained model, which is then corrected using the poetry dataset and uses the cosine similarity scores from the doc2vec model to assess the generated rhyme outputs. The results show that this method ensures good harmony between the output and the given input text.The paper also demonstrated how transformers can learn some poetry styles by exploring poems of some particular poets.

Long Text Generation via Adversarial Training with Leaked Information [9] .Coherent and relevant texts plays a vital role in many domains.Image captioning, dialogue generation and machine translation are some of the examples where long texts are needed. But, with the traditional models, architectures and framework it is difficult to generate coherent texts which are lengthy. Previous works focus on task specific application in supervised domains(Bahdanau,Cho, and Bengio 2014; Vinyals et al. 2015), common non-supervised text generation, which aims to mimic the distribution of real text from a corpus, has received a lot of attention recently. (Graves 2013; Yu et al. 2017; Zhang et al. 2017;Hu et al. 2017).

The approach to train RNN to maximize the log-likelihood of significant words, is prone to risk bias reasons for the discrepancy between training and inference phase. The model sequentially generates the next word based on the words already generated during inference but is self-trained to generate ground truth words.A scheduled sampling approach could be used to solve the problem,but it is experimentally found to be inconsistent.

Generative Adversarial Nets (GAN) (Goodfellow et al. 2014), initially used for continuous data (image generation etc.), when later extended to discrete, sequential data was able to overcome the above problem and has also showed good results. (Yu et al. 2017). One of the main disadvantages of the existing long text generation methods is that the binary guiding signal from the discriminative net (D) is sparse in nature as it is only available when the entire text sample is Generated. The guiding function is also not fully informative.It does not preserve the immediate syntactic structure of the text.So, more guidance should be provided to D to make it more informed.

The problem of sparsity can be resolved by using hierarchy. The output generated text goes through a lot of hierarchies like parts of speech, semantics etc.Thus the generation task can be broken into small sub-tasks in a hierarchical fashion. The LeakGAN approach is used in the paper to address the problems of both sparsity and less informativeness. A hierarchical generator G, consisting of a high-level MANAGER module and a low-level WORKER module are used.MANAGER is a LSTM ,that acts as a mediator.

At each step, this generator obtains the high-level feature representation of D and uses it to create a guiding target for the worker module at that point in time.Since D's information is kept internally and it does not provide such information to G. It is called leakage of information from D.

The WORKER then encodes current generated words with another LSTM and then combines the output of the LSTM and the goal embedding provided earlier to take the final action at the current state.The guiding signals from D in terms of the scalar reward signals is available as a goal embedding vector .This guides how G can be improved.

Extensive experiments on synthetic and real data are performed on the model. For real data, text in EMNLP2017 WMT News, COCO Image Caption and Chinese Poems as the long, mid-length and short text corpus, respectively are used. In all the cases, LeakGAN has shown good results and improvements compared to previous models in terms of BLEU statistics and human Turing test.Deep investigation on the interaction between MANAGER and WORKER, shows that without any supervision LeakGAN learns sentence structures like punctuation, clause structure and long suffix.

Data-to-Text Generation with Content Selection and Planning [10].The work is largely based on non-linguistic input (Reiter and Dell 2000; Gat and Kramer 2018). The input may be in various forms,ranging from databases of records, spreadsheetsexpert systems knowledge bases, simulations of physical systems etc.Traditional methods for data-to-text generation (Kukich 1983; McKeown 1992) implements pipeline modules that include content planning ,sentence planning and surface realization.

The traditional models and architectures are based on encoder-decoder architectures as backbone. These models generate fluent texts.But ,these models are not very good at capturing the sequence plans for long text structures.The traditional models are weak at capturing logical context and sequence for long texts and thus fail to generate long coherent texts.

Approach is made to capture the context and plan to generate coherent long texts. Pointer network is used to capture the effects that seq2seq models are unable to do.The network learns conditional probability and relation between neighboring parts of texts to generate coherent text. Attention mechanism is further used to focus on specific parts of input data

to get the significant parts.

The input is in the form of a table the ROTOWIRE dataset of basketball player.Content selection and planning operates on the input records of a database to produce a content plan . This plan is based taking into consideration the dependencies of other records and thus considering the joint probabilities.

The generated text is long and it follows a canonical structure. The order corresponds to the sequence of entities that appear in the game summary. The text generation phase comprises probability of output text that is conditioned on the content plan and input table.The encoder-decoder model uses attention mechanism to capture more focussed content.

Content plans from the ROTOWIRE game summaries were extracted using information extraction (IE) approach .Ensembling a model of three convolutional models and 3 bidirectional LSTM models are used. It showed good performance.

One-layer pointer networks were used for content planning and two-layer LSTMs were used for text generation ,while BPTT was used for text decoding. The result of the experiment showed improvements in terms of accuracy, BLEU score,precision and recall compared to State of the art (Wiseman et al. (2017)) . Thus,the data-to-text generation model got enhanced when content selection and planning were taken into consideration while generating text. Generation quality improved in terms of the number of relevant facts contained in the output text,as well as the order of presenting them. Further work can be extended to detailed instructional plans that deal with multiple decision-making facts that can be learned and the way they can be identified across future languages and dialects.

2.2 NLG ON INDIAN LANGUAGES

IndicNLG Suite: Multilingual Datasets for Diverse NLG Tasks in Indic Languages [18].The paper focus on five diverse tasks, namely, biography generation using Wikipedia infoboxes (WikiBio), news headline generation, sentence summarization, question generation and paraphrase generation.They describe the process of creating the datasets and present statistics of the dataset, following which we train and report a variety of strong mono- lingual and multilingual baselines that lever- age pre-trained sequence-to-sequence models and analyze the results to understand the chal- lenges involved in Indic language NLG. To the best of our knowledge, this is the first NLG dataset for Indic languages and also the largest multilingual NLG dataset. The methods can also be easily applied to modest- resource languages with reasonable monolingual and parallel corpora, as well as corpora containing structured data like Wikipedia.

This might help research in NLG on diverse languages and tasks, particularly for Indic languagesNLG is the process of generating textual output (Gatt and Krahmer, 2018). Initial work on NLG focused on tabular input (Reiter and Dale, 1997) but in a general setting the input can also belong to one or more modalities such as text, images, videos, audio, etc. NLG progress had been hindered by the lack of data for different tasks and languages but, recently, the increasing availability of large scale datasets (Narayan et al. 2018a; Wiseman et al. 2017; Lebret et al. 2016, inter alia), along with the advancements in neural networks pre trained on large amounts of text (Lewis et al., 2020; Raffel et al., 2020) have led to substantial progress in NLG.

The work motivated the creation of the multilingual extreme summarization(XL-Sum) (Hasan et al., 2021b) dataset spanning 44 languages, as well as the cross-lingual summa-rization dataset (Hasan et al., 2021a).There is no existing or widely used suite of datasets for Indic NLG 3 , this paper aims to fill this gap, via the IndicNLG suite where new datasets for 11 Indic languages are created.This has a spanning over five NLG tasks such as biography generation using Wikipedia infoboxes (WikiBio), news headline generation, sentence summarization, question generation and paraphrase generation. Along with creation of datasets, the paper has also trained a variety of baseline models focusing on pre-training and multilingualism to establish benchmarks which other researchers can consider as a starting point for their NLG research.It is expected that the efforts will also lead to a significant growth in NLG for Indic languages, through additional datasets covering more languages, newer tasks and novel methods targeting language families.

Significant contributions of the paper is to create the IndicNLGSuite, a collection of NLG datasets for five diverse NLG tasks spanning 11 languages from the two major language families (Indo-Aryan and Dravidian) in India.

To show feasibility of mining NLG datasets.It is the largest and linguistically most diverse multilingual NLG dataset, comprising a total of 8.5M examples across 11 languages and 5 tasks (55K to 5.57M examples for a task- language pair), opening up possibilities for multilingual NLG research.Rich statistics on the datasets are provided,to use it along with English counterparts.The paper provides fine-tuned baseline models for 11 Indian languages.

The work has shown clear evidence of the advantage of language family specific pre- trained models compared to language agnostic ones, on the final downstream performance.Chen et al. (2021) propose an NLG benchmark for three languages ‘fr’, ‘de’ and ‘es’ along with ‘en’, for three tasks of story generation, headline generation and question generation. In contrast, the IndicNLG benchmark covers 11 Indic languages and five tasks, making it the first for Indic languages as well as the most linguistically diverse NLG benchmark.The works are used for Headline generation ,sentence summarization,paraphrase generation question generation tasks.

The works created IndicNLP Suite, a collection of datasets for 5 diverse NLG tasks in 11 Indic languages, with the aim of creating much-required standard benchmarks to drive NLG research for Indic languages. In addition, this is the largest multilingual NLG dataset .The methods are simple enough to create similar datasets for modest resourced languages.Future baselines may

consider multilingual fine-tuning as the starting point. Future works may focus on extending the datasets for additional Indic languages, as well as adding new tasks and stronger baseline models.

Poetic Machine: Computational Creativity for Automatic Poetry Generation in Bengali [19].The limitations of lexical metrics for NLG evaluation are well acknowledged - particularly for morphologically rich target languages (Guzmán et al., 2016).Exploration can be done on embedding based metrics (Sellam et al., 2020) to track progress on these tasks.Bengali is a morpho-syntactically rich language and partially phonemic.The design process involves rhythm understanding from the given input and follow-up rhyme generation by leveraging syllable / phonetic mapping and natural language generation techniques.In order to construct a syllabification engine based on grapheme-to-phoneme mapping,to comprehend the rhyme given as input.A poetical rhythmic follow-up statement is formed,using weight-based aggregation and n-gram matching Three criteria were used to assess the quality of the mechanically generated rhymes.Poeticness, grammaticality, and meaningfulness were the three criteria used.During the last two years, there has been a significant increase in research focus on affect understanding, or the second level of cognition.There have only been a few attempts to create poetry by machines.The computational creativity paradigm is still in its infancy, and most of those who have tried it have failed miserably. The field of automatic poetry generation was pioneered by Bailey (1974), although Funkhouser (2009) quotes work going back to the 1950s. These systems were written by actual poets who were keen to explore the potential of using computers in writing poetryand were not fully autonomous. Thereafter, Gervás and his colleagues were the first to discuss sophisticated approaches to automatic poetry generation (Gervás 2000;2001a; 2001b; 2002a; 2002b; Díaz-Agudo, Gervás, and González-Calero 2002; Gervás et al. 2007).Manurung (2004) developed a system that generates metrically limited poetry from a given topic using a grammar-driven formulation.

The main aim was to collaborate with the user rather than trying to have the machine create poetry on its own. And not a full poem, but one line of poetry at a time.

This task is divided into two parts: rhyming comprehension and rhyme generation. Rhyme comprehension includes deciphering the lyrical structure of the input line. The creation of rhymes is based on the use of a Bengali syllabification engine and a rhyme generator.SVM-based classifier for predicting the structure of the output sentence and candidate word creation, along with bigram

pruning and weighted aggregation for selecting the actual words to be utilized in the resulting rhyming line.Finally, the poetry generating model is evaluated in terms of the three dimensions of poeticness, grammaticality, and meaning.The results of some preliminary studies on the automatic generation of Bengali poems are presented in this study. Bengali is a morpho syntactically rich language that borrowed Sanskrit's traits and fundamentals for its poetry. As a result, Bengali rhyme generation is a difficult problem to solve. The method used here is innovative, and it is based on interaction with the user, who enters a line of poetry, which the system attempts to comprehend in order to generate a text line that follows the rules and metres of Bengali poetry and rhymes with the input.

For predicting the structure of the output sentence and for candidate word production, a Bengali syllabification engine and an SVM-based classifier are used, which is based on a notion of semantic importance in terms of proximity mappings generated from ConceptNet translations. Bigram trimming and aggregation are currently used to make the final selection of the genuine poetry words

Although poetry created by humans does not always contain semantically linked terms, using the concept of semantic relevance provides a computationally cheap technique to automatically create meaningful rhymes.

Chapter 3

MODELS

3.1 RNN-Recurrent Neural Network

RNNs are a sort of Neural Network in which the output from the previous step is used as input in the next stage. All of the inputs and outputs in standard neural networks are independent of one another, however in some circumstances, such as when predicting the next word of a phrase, the prior words are necessary, and so the previous words must be remembered. As a result, RNN was created, which used a Hidden Layer to overcome the problem. The Hidden state, which remembers certain information about a sequence, is the most essential aspect of RNN.

RNNs have a "memory" that stores all information about the calculations. It employs the same settings for each input since it produces the same outcome by performing the same task on all inputs or hidden layers. Unlike other neural networks, this decreases the complexity of the parameters.

Recurrent Neural Networks are a Deep Learning technique for modeling sequential data (RNN). Prior to the introduction of attention models, RNNs were the go-to solution for dealing with sequential data. A deep feedforward model (Figure 3.1)may require specific parameters for each member of the sequence. It may also be unable to generalize to sequences of varying lengths.

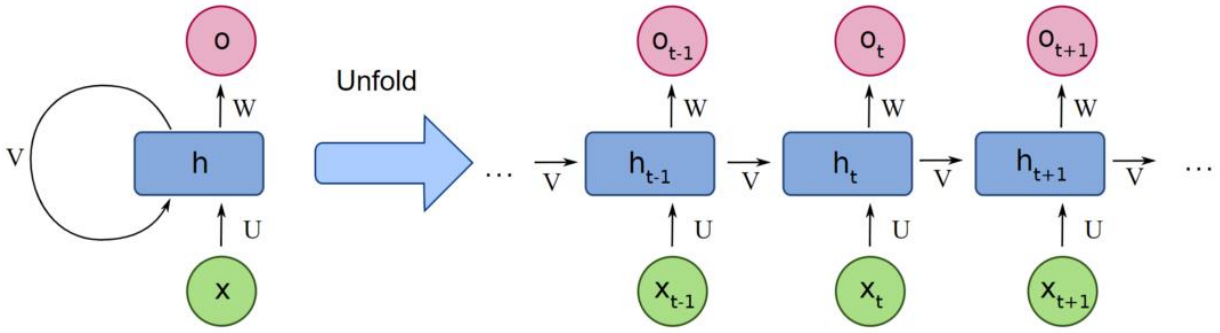

Figure 3.1.A deep feedforward model

Recurrent Neural Networks apply the same weights for each element of the sequence, reducing the number of parameters and allowing the model to generalize to different lengths of sequences. Because of their design, RNNs generalize to structured data other than sequential data, such as geographical or pictorial data.

Like many other deep learning approaches, recurrent neural networks are relatively new. They were first developed in the 1980s, but it wasn't until recently that we realized their full potential.

In the domains of AI, machine learning, and deep learning, neural networks simulate the function of the human brain, allowing computer systems to recognise patterns and solve common problems.

RNNs are a sort of neural network that can be used to model data in a series. The behavior of RNNs, which are built from feedforward networks(Figure 3.2) ,is comparable to that of human brains. Simply said, recurrent neural networks are better at anticipating sequential data than other algorithms.

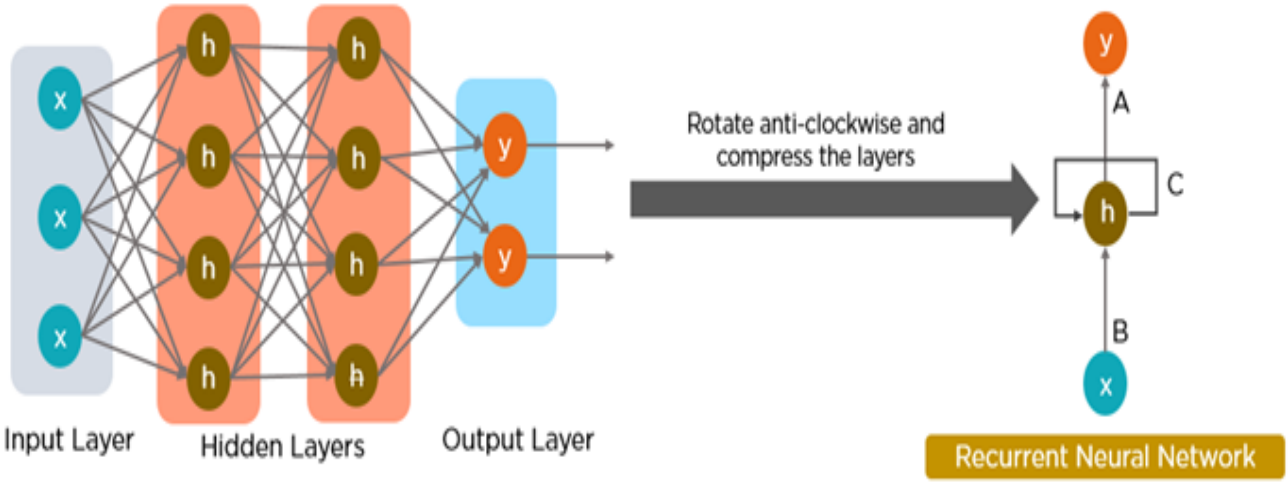

Figure 3.2.RNN with input layer ,hidden layer and output layer.

In normal neural networks, all of the inputs and outputs are independent of one another; but, in some cases, such as when predicting the next word of a phrase, the preceding words are required,

and so the prior words must be remembered.Thus multiple layers are required . As a result, RNN was developed, which solved the problem by using a Hidden Layer. The Hidden state, which remembers specific information about a sequence, is the most significant component of RNN(Figure 2).RNNs have a Memory that contains all of the calculations' data. Because it achieves the same result by completing the same operation on all inputs or hidden layers, it uses the same parameters for each input.

Architecture of RNN

RNNs are a sort of neural network with hidden states and the ability to use previous outputs as inputs(Figure 3.3 and 3.4)

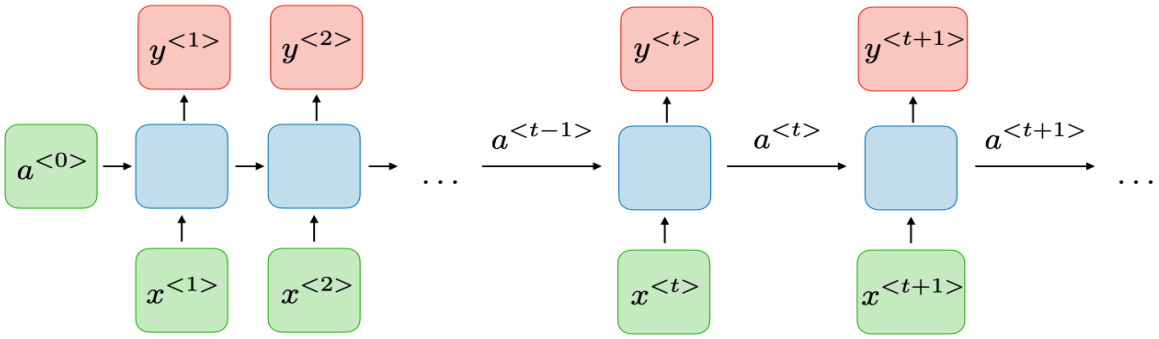

Figure 3.3 Ability to use previous outputs as inputs

For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{and} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

Figure 3.4 Labels of figure 3.3

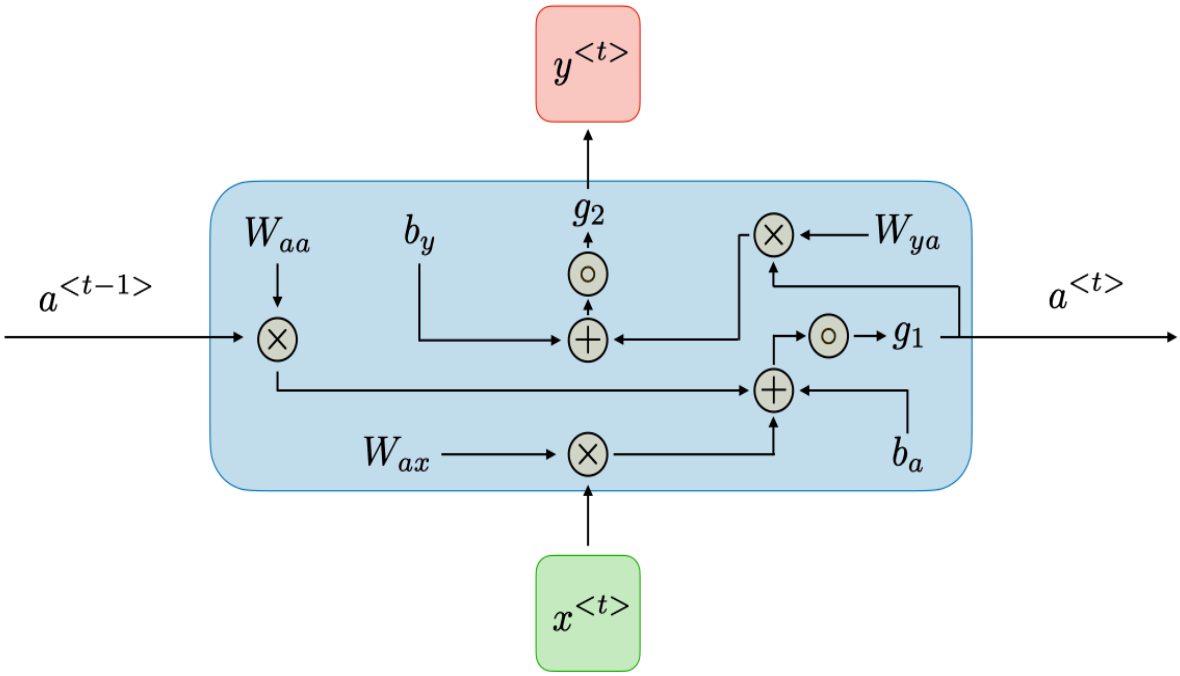

Figure 3.5 RNN shown in a simplified way.

In figure 3.5 a combination of all layers together is shown to diagrammatically represent RNN.

Types of RNN:-One to One: This is a one-on-one situation. Traditional neural networks have a one-to-one architecture.

One to Many: In a one-to-many network, a single input can result in several outputs.In the production of music, for example, there are far too many networks.

Several to One: In this scenario, many inputs from different time steps are combined to produce a single output. Such networks are used in sentiment analysis and emotion identification, in which the class label is decided by a sequence of words.

Many to Many: There are several alternatives for many to many. There are three outputs from two inputs. Many-to-many networks are used by machine translation systems, such as those that translate English to French or vice versa.

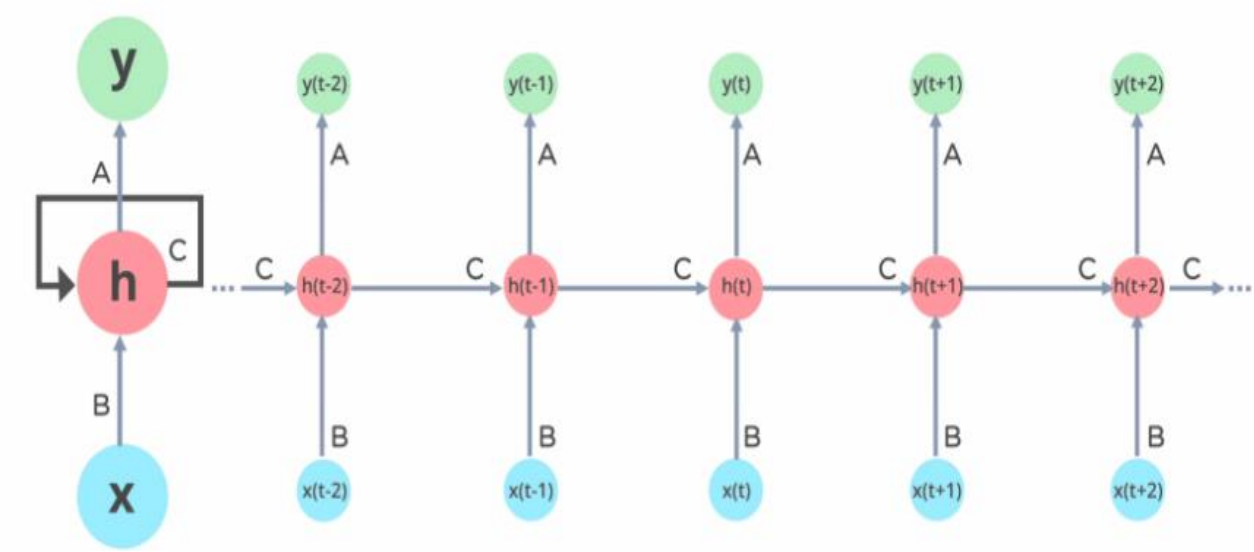

Figure 3.6 Information is cycled in a loop to the middle hidden layer

Working of RNN -In recurrent neural networks, information is cycled in a loop to the middle hidden layer.(Figure 3.6)The neural network's input is received and processed by the input layer x before being passed on to the middle layer.

In the hidden layer h, there are several hidden layers, each with its own activation functions, weights, and biases. A recurrent neural network can be used if the various parameters of different hidden layers are not influenced by the preceding layer, i.e. the neural network has no memory.

The Recurrent Neural Network will standardize the various activation functions, weights, and biases, guaranteeing that each hidden layer has the same properties. Rather than creating several hidden layers, it will simply generate one and loop over it as many times as needed.

Problems with RNN: Exploding gradients problem and Vanishing gradient problem

Exploding gradients arise when the algorithm assigns the weights an abnormally high priority for no obvious reason. Fortunately, truncating or squashing the gradients is a straightforward approach.

When the gradient values are too modest, the model stops learning or takes far too long. This was a major problem in the 1990s, and it was significantly more difficult to solve than the expanding gradients. Fortunately, the LSTM idea developed by Sepp Hochreiter and Juergen Schmidhuber solved the problem.

Application of RNN-Recurrent Neural Networks are utilized to solve a wide range of sequence data challenges. There are several sequence data types, but the following are the most common: Audio, text, video, and biological sequences are all possible. Example: Speech recognition,Generation of music,Automated Translations,Analysis of video action,Sequence study of the genome and DNA etc.

Brief conclusion of RNN-Recurrent Neural Networks are a flexible tool that may be used to a wide range of problems. They are used in a number of strategies for language modeling and text generation. They are also used in speech recognition.

RNN's struggle to learn long-term dependencies, which implies they don't understand linkages between facts that are separated by several stages.

3.2 LSTM-LONG SHORT TERM MEMORY

LSTM networks were created primarily to address the issue of long-term dependency that recurrent neural networks RNNs suffer (due to the vanishing gradient problem). LSTMs vary from standard feedforward neural networks in that they feature feedback connections.

This trait enables LSTMs to process complete data sequences (e.g., time series) without considering each point in the sequence individually, but rather by preserving important information about earlier data points in the sequence to aid in the processing of new data points. As a result, LSTMs excel in processing data sequences such as text, audio, and time series.

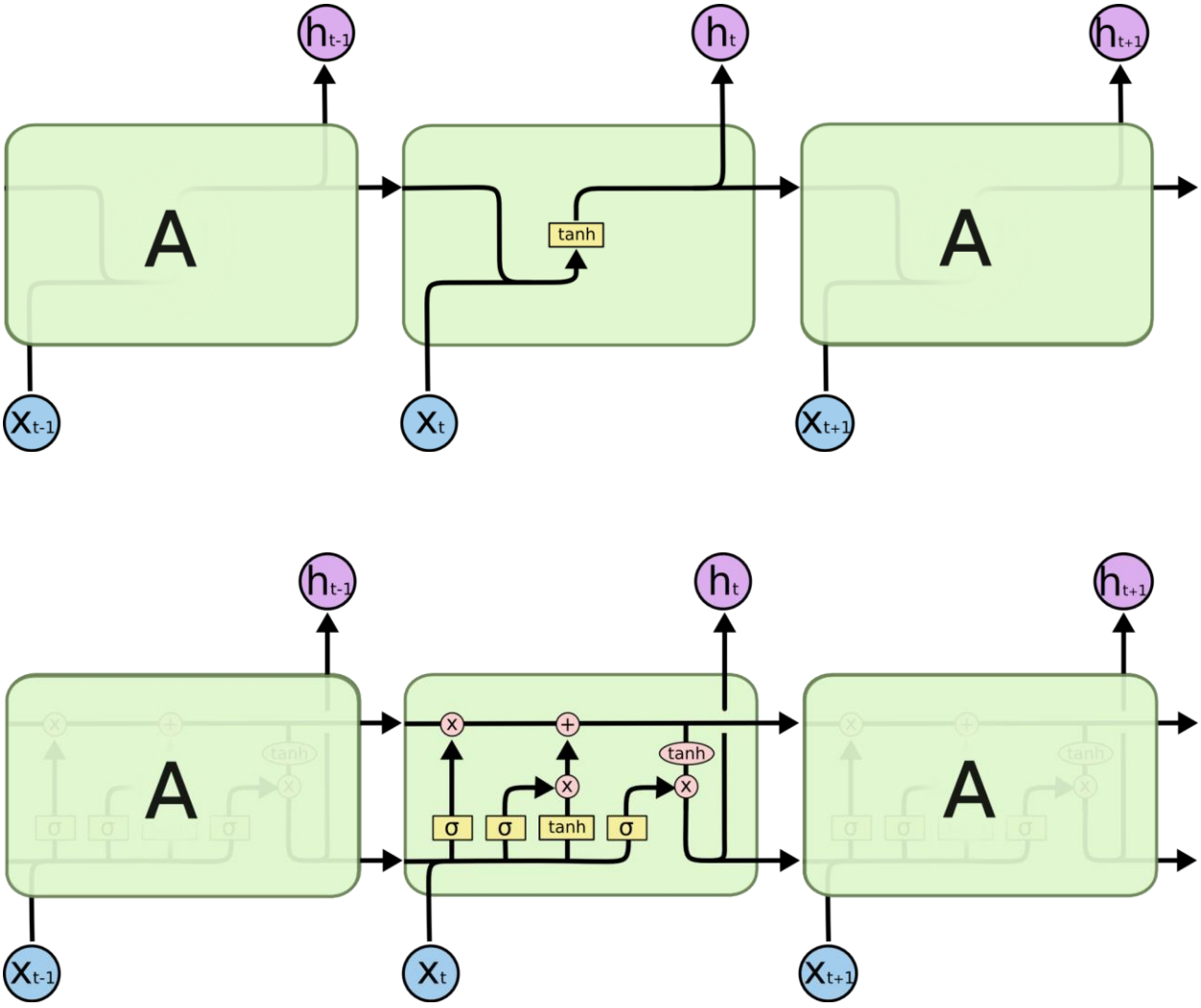

Figure 3.7 The repeating module has 4 interacting layers and transports a full vector from one node's output to the inputs of another node

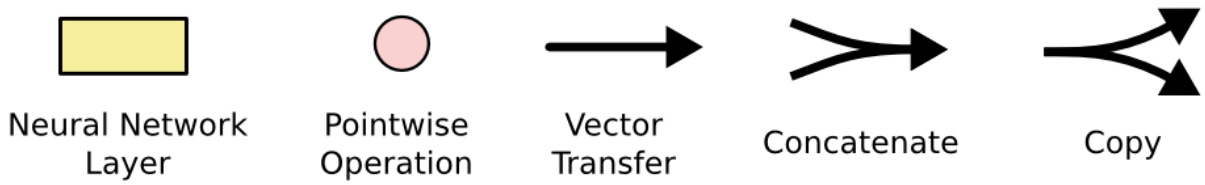

Figure 3.8 Different components

Each line in the figure 3.7 and 3.8 transports a full vector from one node's output to the inputs of another node .

Pink circles indicate pointwise operations such as vector addition, whereas yellow boxes represent trained neural network layers. Lines that merge suggest concatenation, lines that fork denote their content being copied and the copies being sent to distinct destinations.

The cell state (Figure 3.9)is similar to a conveyor belt. It follows the whole chain, with only a few small linear interactions. It is quite simple for information to just pass over it unmodified.

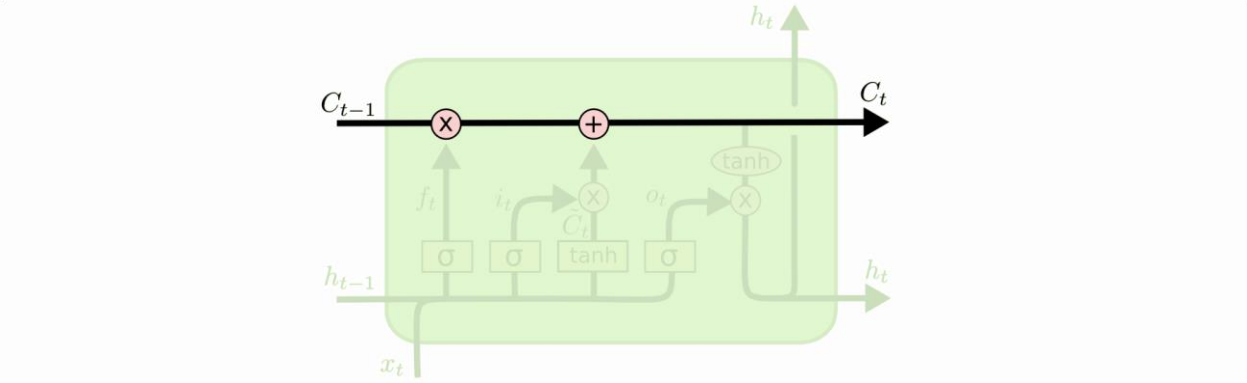

Figure 3.9 LSTM cell state

The LSTM may delete or add information to the cell state, which is carefully controlled by structures known as gates.

Gates are a method of optionally allowing information to pass through. They are built with a sigmoid neural net layer and pointwise multiplication(Figure 3.10).

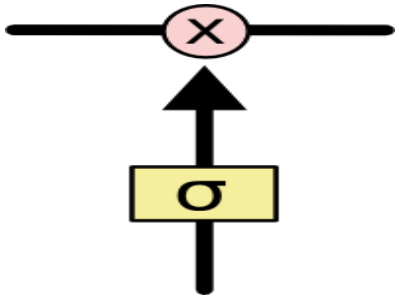

Figure 3.10 Gate of LSTM

The sigmoid layer produces integers ranging from zero to one, indicating how much of each component should be allowed through. A value of zero indicates "let nothing through," whereas a value of one indicates "allow everything through.Three of these gates are present in an LSTM to safeguard and govern the cell state.

Architecture of LSTM-The primary distinction between RNN and LSTM designs is that the hidden layer of an LSTM is a gated unit or gated cell. It is made up of four layers that interact with one another to create the cell's output as well as the cell state. These two items are subsequently passed on to the following concealed layer. Unlike RNNs, which only have one neural net layer of tanh, LSTMs feature three logistic sigmoid gates and one tanh layer. Gates have been designed to limit the amount of information that passes through the cell. They decide which information will be needed by the next cell and which will be rejected.The output is normally in the range of 0-1, where '0' means reject all' and '1' means 'include all.

Application of LSTM-Before being used in real-world applications, LSTM models must be trained with a training dataset. The following are some of the most demanding applications:Language modeling or text creation is the calculation of words when given a series of words as input. Language models can be used at the character, n-gram, phrase, and even paragraph levels.Image processing is the process of analyzing an image and converting the results into a text. This necessitates the availability of a dataset including a large number of images with meaningful

descriptions.Speech and Handwriting Recognition,Music generation is analogous to text generation in that LSTMs anticipate musical notes rather than text by studying a combination of provided notes fed as input,Language Translation is the process of transferring a sequence in one language to a sequence in another. A dataset including phrases and their translations is first cleaned, similar to image processing, and just a portion of it is utilized to train the model. An encoder-decoder LSTM model is used, which transforms the input sequence to its vector representation (encoding) before translating it.LSTM could overcome the problem of disappearing gradients, so LSTM became popular. However, they fail to totally eradicate it. The issue is that the data must still be sent from cell to cell for analysis. Furthermore, with the inclusion of new features (such as forget gates), the cell has grown rather sophisticated.Because of the linear layers present in each cell, LSTMs require high memory bandwidth, which the system often can not offer. As a result, LSTMs are relatively wasteful in terms of hardware.

LSTMs are impacted by distinct random weight initializations and so behave similarly to feed-forward neural nets. Instead, they choose minimal weight initialization.LSTMs are prone to overfitting, and the dropout strategy is difficult to use to mitigate this problem. Dropout is a regularization strategy in which input and recurrent connections to LSTM units are probabilistically omitted from activation and weight updates during training a network.

3.3 GPT 2

GPT-2 is a self-supervised transformers model pre trained on a very large corpus of English data. This implies it was trained on raw texts solely, with no human labeling (thus its ability to handle a large amount of publically available data), and then used an automated procedure to build inputs and labels from those texts. It was specifically taught to guess the next word in sentences.

The GPT design leverages attention instead of earlier recurrence- and convolution-based architectures to create a deep neural network, especially a transformer model. The model's attention processes allow it to choose to focus on parts of incoming text that it anticipates will be the most relevant. This model significantly improves parallelization and outperforms earlier benchmarks for RNN/CNN/LSTM-based models.

GPT-2 uses Byte pair encoding, Byte pair encoding, also known as digram coding, is a basic type of data compression in which the most common pair of successive bytes of data is replaced with a

byte that does not appear in that data. A vector is generated for every letter.To recreate the original data, a table of replacements is necessary.A vector is generated for every letter.

Chapter 4

Experiments and Results

4.1 Data set

To generate poems in Bengali the different models are trained with bengali corpus.The details about the dataset are given below.The dataset comprises poems of two eminent Bengali poets Nobel laureate Rabindranath Tagore and Sukanto Bhattacharya.The Rabindranath corpus comprises 1532 files in txt format. Each file is a poem in Bengali.The Sukanto corpus comprises 119 poems in txt format. Each file is a poem in Bengali.(Table 4.1).The average word count per poem is given in the table below.

Name of Corpus	Average word count	Total number of files/poems
Rabindranath Tagore Corpus	233	1532
Sukanto Bhattacharya Corpus	177	119

Table 4.1 Dataset information

4.2 Experimental Setup

The individual corpus are appended together to obtain 2 separate merged corpus. One for Rabindranath Corpus and one for Sukanto Corpus.These 2 separate corpus are individually trained on RNN .There are 50 epochs to train the RNN in Google colab.Training is done on a default batch size of 128.After training the model with the corpus, it is able to generate poetic texts by giving a set of words as input.The generate function takes two parameters: the set of words and number of words to be generated in the poem.The output is the set of words given as input along with words after it according to the length given.The LSTM model works by predicting the next word.Training has been done on two individual corpus separately.One hot encoding has been used to train the

model about the words.Word level training is done in LSTM so on being trained, it can generate words from the vocabulary it has learnt.It works by predicting the next word. The generating function takes a set of words as input and length of text in this case poem to be generated.The model word by predicting the next word.So now the model has 1 more word than the input,now this goes into the generating function again to predict the next word.This method has been used repeatedly to generate the following words.

In GPT-2 the two individual corpus are fine-tuned in the GPT-2 separately.

After fine-tuning,the model is given a set of words as input and a length as a parameter to the generating function.The model can then generate words following the input words to generate the entire poem upto the given length.

GPT-2 training with poems and keywords.Preprocessing the two corpus in a different way has given a better output, compared to the above mentioned ways.In this case the model gets to learn about the context of the words ,as a result the generated output has more meaning.

4.3 Data preparation with keyword extraction

Tf-idf:In information retrieval, tf–idf (also TF*IDF, TFIDF, TF–IDF, or Tf–idf) is a numerical statistic that is supposed to indicate how essential a word is to a document in a collection or corpus. It is frequently employed as a weighting factor in searches for information retrieval, text mining, and user modeling.

The tf–idf value grows according to the number of times a word appears in the document and is offset by the number of documents in the corpus that include the word, which helps to account for the fact that some words appear more frequently in general. One of the most often used term-weighting techniques currently is tf–idf. According to a 2015 survey, 83 percent of text-based recommender systems in the digital libraries use tf–idf.

Term frequency:

The term frequency, tf(t,d), is the frequency of occurrence of term t inside document d,

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

where $f_{t,d}$ is the raw count of a term in a document, i.e., the number of times that term *t* occurs in document *d*. Note the denominator is simply the total number of terms in document *d* (counting each occurrence of the same term separately).

Inverse Document Frequency:

The inverse document frequency indicates how much information a word gives, i.e., whether it is frequent or uncommon across all texts. It is the logarithmically scaled inverse fraction of documents containing the phrase (obtained by dividing the total number of documents by the number of documents containing the term and then calculating the logarithm of that quotient):

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Here,

N=Total number of documents in the corpus |D|

$$|\{d \in D : t \in d\}|$$

number of documents where the term t appears (i.e. tf(t,d) is not 0) , If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to

$$1 + |\{d \in D : t \in d\}|$$

Calculation of tf-idf

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

A high weight in tf–idf is achieved by a high term frequency (in the given document) and a low term frequency in the entire collection of documents; the weights therefore tend to filter out common words. The value of idf (and tf–idf) is larger than or equal to 0 since the ratio within the

idf's log function is always greater than or equal to 1. The ratio inside the logarithm approaches one when a phrase appears in more documents, bringing the idf and tf–idf closer to zero.

The dataset obtained from Sukanto corpus is small and the average size of poems are not very much different from each other.Tf-idf has been used to extract keywords from each individual poem.the threshold has been kept as 0.15 to get keywords from each poem.Whereas, the Rabindranath corpus poem sizes vary from one another and thus tf-idf value is kept lower 0.05.

The problem with Tf-idf is that it does not give information about the significance in terms of context of a given word.As in a poem the context of a word plays an important role,so to incorporate the context in the extracted words ,TF-idf is used in a modified way.

Tagging of each keyword has been done separately.The tags consist of the nth word it is in the poem,whether the word is the starting word of a poem.Whether the keyword is an ending word of the poem.Whether the keyword is a starting word of a sentence in the poem, or ending in a poem.

Whether the keyword is a starting word of a paragraph in the poem or ending word of paragraph in the poem.After making all the taggings of the keyword the information about the context of the word has been added .The keyword extraction worked at word level so all the words carried meaning.

Other method of keyword extraction have also been used like BERT tokenizer,but in Bengali data it has issues.In English it shows good results and additional context information need not be added to it.But the Bengali ‘juktakkhor’ splitting takes place because of working on character level.So Tf-idf with tagging has been used for preparing the keywords for each individual poem.

Preparation of data to train the GPT-2

Keyword of i-th poem has been placed with a separator tag with the i-th poem,thus, the dataset has tagged keyword followed by separator followed by that i-th poem.

This was the preparation of the dataset.This process has been applied on each individual corpus to generate the datasets for fine-tuning in GPT-2 .On being fine-tuned, the model learns the keywords and the poem corresponding to the keywords.The generating function now acts in a different way by taking words as input and generating poems as output,but the words do not come

in front,rather the generated poem has the words in it ,thus the generated output carries more meaning.

4.4 Results

Evaluation has been done by human beings.Human based evaluation has been adopted in this.The different models output are generated on two different corpus and rating has been done . 3 different individuals evaluate by rating the poems generated by each model according to 3 criteria on a scale of 1 to 5 for context or meaning where, 1 poor ,and 3 average 5 very good .Ratings are taken for each poem from each model by 3 different individuals .The ratings are tabulated in Tables 4.2 to 4.7..To detect style of writing, the observers are given to identify whether the observer is able to identify the poetic style of the generated corpus or not. 0- not able to detect 1-able to detect.

Human Evaluation on Sukanto Corpus

OBSERVER 1	RNN	LSTM	GPT-2	GPT-2 with keywords
Detect style	0	1	0	1
Meaning or context	2	3	3	3.5

Table 4.2 Ratings of 1st Human Evaluator on Sukanto Corpus

OBSERVER 2	RNN	LSTM	GPT-2	GPT-2 with keywords
Detect style	1	1	1	1
Meaning or context	2	2	2	3

Table 4.3 Ratings of 2nd Human Evaluator on Sukanto Corpus

OBSERVER 3	RNN	LSTM	GPT-2	GPT-2 with keywords
Detect style	1	1	1	1
Meaning or context	2.5	3	2.5	2.5

Table 4.4 Ratings of 3rd Human Evaluator on Sukanto Corpus

Human Evaluation on Robindronath Corpus

OBSERVER 1	RNN	LSTM	GPT-2	GPT-2 with keywords
Detect style	1	1	1	1
Meaning or context	3	2	2	3

Table 4.5 Ratings of 1st Human Evaluator on Robindronath Corpus

OBSERVER 2	RNN	LSTM	GPT-2	GPT-2 with keywords
Detect style	1	1	0	1
Meaning or context	3	3	3	3.5

Table 4.6 Ratings of 2nd Human evaluator on Robindronath Corpus

OBSERVER 3	RNN	LSTM	GPT-2	GPT-2 with keywords
Detect style	1	1	1	1
Meaning or context	2	3	2	3

Table 4.7 Ratings of 3rd Human evaluator on Robindronath Corpus

4.5 Analysis

The ratings of different individuals on different corpus has been tabulated above.Different Human Evaluators have different poetic tastes, and accordingly the keywords or set of words have been given to generate poems. Some words that are not well known to the models do not produce proper results as training has not been done well on these words,while in other words the models act differently.Especially in case of poem generation with keywords,this has been observed to a good extent.The keywords that were less common to the poems could not be used to teach context of the words in the poems,to the models while some other words could even generate synonyms, replacing the actual word.Thus adding context showed better result than plainly training in these cases.While in some cases with less frequent words the generated output was not that good.

Chapter 5

Conclusion and Future Work

Natural language generation is a very challenging task,especially when it comes to poems. Different models have been explored for the way poems are generated by them in Bengali language ,and how by adding context using tf-idf with tagging can be used to generate poems from keywords have been observed.

Future work can be done to explore more ways to generate poems.The work can be extended for lyrics generation.,it is a challenging task to generate poems in low resource language like Bengali as pre-trained models for Bengali text generation are coming up new, and not vastly available.To capture the rhymes of poems classifiers like svm can be explored.The comparison based on different models have been done and exploration can be done further to improve the quality of generated text

REFERENCES

[1] Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H. and Jiang, M., 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys (CSUR)*.

[2] Gatt, A. and Krahmer, E., 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, pp.65-170.

[3] Oliveira, H.G., 2017, September. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th international conference on natural language generation* (pp. 11-20).

[4] Zheng, H.T., Wang, W., Chen, W. and Sangaiah, A.K., 2017. Automatic generation of news comments based on gated attention neural networks. *IEEE Access*, *6*, pp.702-710.

[5] Liu, Q., Zou, L., Che, H., Wang, H., Jin, Y. and Yang, H., 2017, June. A Creative Computing Based Inspiration Assistant to Poem Generation. In *2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)* (pp. 469-476). IEEE.

[6] Yang, H. and Hugill, A., 2013. The creative turn: new challenges for computing. *International Journal of Creative Computing*, *1*(1), pp.4-19.

[7]Cropley, A., 2006. In praise of convergent thinking. *Creativity research journal*, *18*(3), pp.391-404. [8] Santillan, M.C. and Azcarraga, A.P., 2020, July. Poem generation using transformers and Doc2Vec embeddings. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.

[9Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y. and Wang, J., 2018, April. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

[10] Puduppully, R., Dong, L. and Lapata, M., 2019, July. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 6908-6915).

[11] Mann, W.C., 1983, August. An overview of the Penman text generation system. In *AAAI* (pp. 261-265).

[12Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D. and Yan, R., 2019, July. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, No. 01, pp. 7378-7385).

[13] Onodera, K., Akimoto, T. and Ogata, T., 2012. A state-event transformation mechanism for generating micro structures of story in an integrated narrative generation system. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 34, No. 34).

[14Manurung, H.M., 2003. An evolutionary algorithm approach to poetry generation [Ph. D. Thesis]. *Edinburgh: University of Edinburgh*.

[15] Das, A. and Gambäck, B., 2014. Poetic Machine: Computational Creativity for Automatic Poetry Generation in Bengali. In *ICCC* (pp. 230-238).

[16] Aurobindo, S., 2004. *Letters on poetry and art*. Sri Aurobindo Ashram Publication Department.

[17] Bailey, R.W., 1974. Computer-assisted poetry: the writing machine is for everybody. *Computers in the Humanities*, pp.283-295.

[18Kumar, A., Shrotriya, H., Sahu, P., Dabre, R., Puduppully, R., Kunchukuttan, A., Mishra, A., Khapra, M.M. and Kumar, P., 2022. Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages. *arXiv preprint arXiv:2203.05437*.

[19Das, A. and Gambäck, B., 2014. Poetic Machine: Computational Creativity for Automatic Poetry Generation in Bengali. In *ICCC* (pp. 230-238).

[20]https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[21]https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-netwoks-rnn/

[23] https://www.quora.com/What-is-LSTM

[24]https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[25]https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[26]https://jalammar.github.io/illustrated-gpt2/

[27]https://towardsdatascience.com/conditional-text-generation-by-fine-tuning-gpt-2-11c1a9fc639d