A Survey of Text Generation Models

Mayuresh Muley
Student, Department of Computer Science
MIT World Peace University, Pune

Pruthav Sanwatsarkar
Student, Department of Computer Science
MIT World Peace University, Pune

Vrushali Kulkarni, PhD
HOS, Department of Computer Science
MIT World Peace University, Pune

ABSTRACT
Natural language processing (NLP)is a field of linguistics and computer science which focuses on the interaction of humans and computers. The main aim of natural language processing is to make sure that a computer can understand what a human says and possibly get key insights from the auditory data. Natural language text production is a well-known sub-part of NLP which focuses on converting auditory data from spoken languages into text.

This survey aims to shed some light on crucial details about the past, present and the future of text production algorithms along with an aim to provide a comprehensive overview of how different machine learning techniques are being investigated and studied for different NLP applications. Finally, some important research gaps which were found out through the review are highlighted as the study is drawn to a close. This study also aims to synthesize a guide for beginners in this field and to point them towards related research and popular practices.

Keywords
Text generation, Markov chains, LSTM, NLP, Deep learning techniques

1. INTRODUCTION
One of the most significant and difficult problems that has emerged in the NLP field is that of text generation (also known as natural language generation). In order to generate text which can satisfy basic communication conditions and linguistics, machine learning and artificial intelligence is used. This is called text generation. Over the years, researchers have deduced numerous strategies for a wide range of text generation applications. For eg. Text in search engines is autocorrected and completed using Natural language generation (NLG). Now-a-days translation tools are one the most important applications which use NLG.

With an aim to keep the survey organized and informative, it has been divided it into 4 sections based on NLG techniques:

1. Traditional approaches that were developed in the past (Section 2). These include the recent rise of Deep Learning (DL), Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN),Graph Neural Networks, etc.along with a thorough review of relevant research papers.
2. Overview of approaches which are being currently used and researched (Section 3)
3. A brief description of evaluation metrics which are commonly used to evaluate text generation models (Section 4)
4. Some drawbacks of currently used text generation models and emerging topics in text generation which can be explored further (Section 5).

2. LITERATURE REVIEW
As observed throughout all these years, a lot of work has been done in this field but more is needed to make it useful. Different researchers are developing new algorithms and techniques as well as new models to improve current results of text generations. Some of them will be discussed in this section.

Over the years a lot of improvement in the field of NLG has been seen, and even today, researchers are involved in making NLG applications more useful. Using advanced scientific techniques, algorithms and newer machine learning models, text generation results are improving rapidly. These improvements will be discussed in the following section

Akkaradamrongrat et al. [1] discusses the very important problem of how imbalance in the data leads to classification models becoming biased. Akkaradamrongrat et al. majorly discusses two text generation methods and compares them. These methods are namely Markov Chains and Long Short-term Memory (LSTM). The basis of the paper was to generate advertisements for 3 different industries viz. Cosmetics, Electricity and Sanitary and then classify the impact of the advertisement using a classifier. Markov chains produced text which made less logical sense when interpreted by a human as compared to LSTM which made much better and sensible text data. However, surprisingly the balanced dataset generated by Markow models gave much better accuracy and precision when used for classification as compared to the balanced dataset generated by LSTM. In conclusion, Akkaradamrongrat et al. concludes that Markov Chains technique outperformed the traditional approach of over-sampling and text generation using LSTM in the majority of the models.

Shuohua Zhou [3] discusses different applications of deep learning in text generation by investigating numerous neural network-based text creation methods. Shuohua Zhou [3] performs research on the text summarization tasks involved during text generation and develops a summary generation approach based on improved cluster search and conducts testing. Shuohua Zhou in 2020 [3] discussed the application of deep learning in text generation in their paper. It investigates several deep neural network-based text creation methods, performs research on the text summary task, develops a generative summary generation approach based on improved cluster search, and conducts testing.

Lei Sha et al. [6] describes sequence-by-sequence technique which models table content and structure using a local and global addressing scheme. Here, the local addressing scheme determines which words in the table should be in focus, particularly while generating the description. On the other hand, the Global addressing scheme focuses on getting a particular word for generating the summary. Lei Sha et al. mentions a unique encoder-decoder framework that uses short-term memory. The first step in the sequence starts with

encoding the field values into the table. Then the LSTM decoder generates a summary of the encoded table in natural language. Lei Sha et al. introduce a new dual-attention mechanism phase in the decoding phase which consists of 2 parts viz. World level attention and field level attention where the former is associated with local addressing and the latter for global addressing.

In Preksha Nema et al.[9] text generation modelsare used to generate a natural language summary from the structured data for the specific characteristics of the problem address . It proposes a neural component to address the "stay on" and "never look back" behavior decode. And enters a record for French and German and gives the current record result. Provides improved performance on the target domain.

Junyi Li and Tianyi Tang [10] discusses pretrained language models. Junyi Li and Tianyi Tang also provides an outline about the evolution of PLMs for text generation and depict various basic PLM topologies which are used for text generation. A summary of some essential PLM fine tuning strategies has also been provided.

Sheikh Abujar et al. [11] has provided important insights through their paper about generating Bengali text using bidirectional recurrent neural networks (RNN).

The study shows that bidirectional RNN helps to provide relatively accurate output and better yield. This study also conclusively shows that BRNNs can be used to generate text not only in Bengali but any other language.

Zhiqiang Ma et al. [12] has proposed a new model called context-aware variational auto-encoder. By taking previous data into account this model can predict the next word. This model is based upon context and doesn't take into account what the next words would be. This model consists of a combination of variational auto-encoding and bidirectional LSTM for natural language generation. Zhiqiang Ma et al. [12] also shows that this new model is better than that of Seq2Seq and MASS. Zhiqiang Ma and his team in [12] have proposed a new model called context-aware variational auto-encoder. This model predicts the next word and takes into account previous information. This model is associated with the context and does not take the next words into account. This model combines variational auto-encoder and bidirectional LSTM for natural language generation

Their results also show that this model is better than that of Seq2Seq and MASS.

Chenhan et al. [4] proposes a new text generation method called User Defined Generative Adversarial Networks (UDGAN). The author of the paper [4] states GANs, when used for generating sentences inclined towards a particular sentiment, have to retrained each time a new requirement has to be satisfied. This tends to slow down users which, on the contrary, favor faster model outputs. On closer inspection, it can be observed that the general semantics of the text to be generated remains the same regardless of the required sentiment. Based on this understanding Chenhan et al. propose a model where 2 different discriminators are introduced in GAN where one of them is the discriminator-general, (i.e., the discriminator that is trained to understand and identify semantics and structure of the input) while the other is the discriminator-special which makes sure that the output generated is user-defined.

3. OVERVIEW ON TEXTGENERATION TECHNIQUES

3.1 Text Generation Using Markov Chains

A Markov Chain is a stochastic process that models a finite set of states, where the conditional probabilities of moving from a given state to another is fixed.

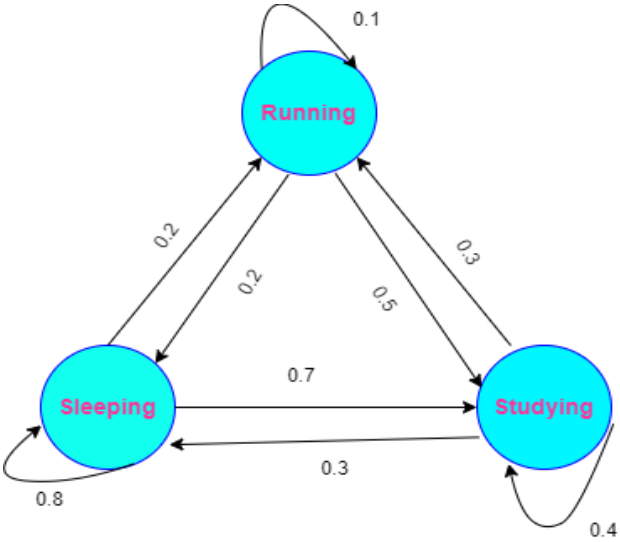

Figure 1: Transition probability of terms

The probability of the current transition is independent of the previous transition. The start stage may be chosen particularly or randomly selected. InFigure 1, the probability of running after sleeping is 60% whereas sleeping after running is just 10%.The important feature to keep in mind here is that the next state is entirely dependent on the previous state, and so Markov chains are memoryless.

In terms of text generation, Markov Chains generates text with random walking.Transition probability is defined by the probability of the occurrence of term when previous term is given.Transition probabilities of terms can be learned by exploring all text in the corpus.

The text can be generated by taking the start term which can be chosen randomly or particularly, the weight of each term can be determined by probability of the occurrence of term in the corpus.The next terms can be generated by weighted random select.

3.2 Text Generation using Recurrent Neural Networks(RNNs)

For NLP using RNN, can be a very powerful tool especially when data that is sequential in nature has to be modeled. It can also be used to develop generative models. This means that, in contrast to forecasting, they can also learn the sequences of a problem and then generate entirely new sequences for the problem domain. Since RNNs contain internal memory, it keeps the memory of the sequence that appears before in order to guess what's coming next in the sequence. This, in principle, makes the network more able to adapt to data that depend on the previous data. The fact that the output which is generated is dependent on the results of the previous time steps makes it highly capable of successfully completing tasks like language generation, language translation, sentiment analysis, etc. This model is used for generating text because of

its sequence modelling capability. But due to this the training on RNN's becomes difficult, and these training issues are long term dependencies. This happens when the algorithm gives high results to weights due to which the model learns nothing resulting in vanishing gradient. To solve these issues various advancements in RNN's have been made in recent years.

3.3 Text Generation using LSTM and GRU

Long Short-Term Memory (LSTM) inherits a similar architecture without the hidden state. LSTMs use the concept of cells which are memory units used to hold the combination of the current input and the previous state as input. LSTM cells are the decision-making unit which decide how memory should be utilized and what should be eliminated. It is also applied to generate synthetic text by learning to predict the next term when the previous sequence of words is given to address the problem of vanishing gradient, LSTM models can be used. The network can make adjustments in the flow of information using additional states in LSTMs called 'cell state'. These additional states give the model the advantage of remembering and forgetting learning selectively.

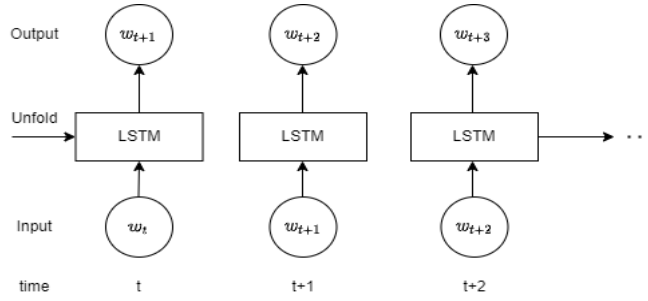

Figure 2: Term prediction process for LSTM

As seen inFigure 2., different layers that can be used in the LSTM model are depicted:

1. Input Layer: This layer takes the input as a sequence of words.
2. Dropout Layer:This layer is an optional layer. This layer is included to get results regularized by randomly turning off activation of some neurons in the LSTM layer.
3. LSTM Layer: Computes the output using LSTM units.
4. Output Layer: Computes the probability of the best possible next word as output

Gated Recurrent Unit(GRU) is another extension of a standard RNN. In the LSTM design, this addition adds a gating network. This modification produces signals that govern previous memory and present information in order to adjust current activation and network state.

This is simpler than the LSTM for which parameter updating is also required for gates.

3.4 Text Generation using Bidirectional RNNs (BRNNs)

One drawback of RNNs can be that future elements in the sequence cannot be taken into consideration while giving output. With the help of BRNNs the output time-step-t can vary on both previous as well as future elements of the sequence. These are simply composed of two independent RNNs, one is forward and another is Backward, both are the opposite direction as shown in Figure 3.

The input to the first RNN is given in normal time order while in the second one as reverse time order. Backward as well as forward information about the sequence can be exchanged at each time step with this type of structure.

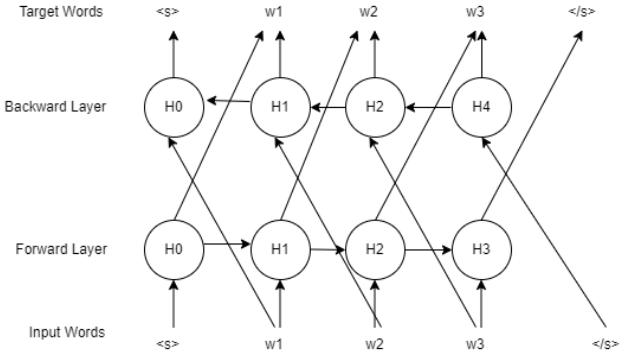

Figure 3:Bidirectional Recurrent Neural Network

3.5 Text Generation Using Variational Auto-Encoders (VAEs)

VAE is one of the powerful deep generative models that works on unlabeled data. Generally, most of the data is unstructured or unlabeled and generative models require a large amount of structured data for training. So VAEs make use of unsupervised data to train the model. It finds the probability distribution of data on given latent distribution. A common autoencoder learns a function which does not train autoencoder to generate images from a particular distribution. While attempting to create a generative model using autoencoders, data generation for input is not the primary objective. It is preferred that VAEs generate output data with some variations which will mostly likely look like the input data.

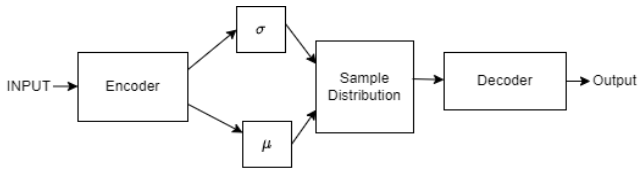

Figure 4: Variational Autoencoder Model

The loss function which forces the model to learn the rich representation of latent space can be broadly defined as the sum of two terms:

Loss Function = Reconstruction Loss + Regularization Term.

The term reconstruction in the preceding definition refers to the mean squared error between the output and input data. The regularisation term has been put here to reduce the distance between latent distribution and some prior distribution to its minimum. A well-known way for unsupervised learning of complex distributions has been developed by VAEs. KL collapse (the situation when the decoder becomes more powerful than the training target and results in solutions with wrong strategy) is one of the primary problems with using VAEs for text generation. In this problem the output of the decoder is generated regardless of latent space. To counter this problem a lot of diverse approaches have been suggested. Among these suggestions is a recent technique for text

generation and addressing KL collapse. This technique proposes that the RNN decoder should be replaced with a dilated CNN which simplifies the control of contextual capacity by changing dilation.

3.6 Text Generation using Generative Adversarial Networks (GANs)

A popular algorithm in Deep learning which takes a different approach than traditional neural networks is Generative Adversarial Networks(GANs).

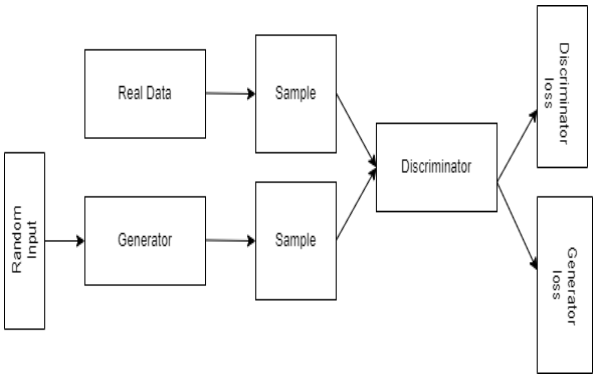

Figure 5: Block Diagram of GANs

The basic idea of GANs is to build a game between two players, that is, a generator and a discriminator. The generator's job is to generate samples that appear to come from the actual distribution (approximated by the training data distribution). The discriminator typically performs supervised learning using a deep network to learn to distinguish true samples from false ones.

During GAN training, the discriminator as well as generator are trained sequentially. This allows us to enhance both the generator and the discriminator while also obtaining human-quality samples from the generator. It's actively being researched[7] how GANs can be utilized in text generation models and how issues related to it can be addressed/ mitigated.

In a bid to reduce exposure bias caused by reinforcement learning, research [8] suggests using adversarial network-based training methods but tends to suffer from exploding or vanishing gradient.

3.7 Text Generation using Pre-Trained language models (PLMs)

The language model is provided a huge amount of unannotated data to begin with and can be fine-tuned on downstream generating jobs. PLMs pre-trained in a big corpus encode substantial language and global knowledge in a huge number of parameters that can increase language understanding and generation quality. Some PLMs use the typical Transformer design based on the fundamental encoder-decoder framework for text creation activities, while others use a decoder-only Transformer.A few examples of PLMs are BERT, GPT, T5 etc.

3.7.1 Encoder-decoder Transformer

A conventional transformer employs the encoder-decoder design, which is comprised of two transformer block stacks. The encoder receives an input sequence, and the decoder attempts to generate an output sequence depending on the mechanism. self-awareness of codecs. Eg.: BERT, T5.

3.7.2 Decoder-only Transformer

Just like many other language modelling architectures, this model uses a single Transformer decoder block. These models ensure that each tile can only pay attention to the previous tiles by applying one-way self-attention masking. Eg.: CTRL, GPT

Seq2seq masking is a logical technique for decoder-only Pre trained language models to perform conditional generation tasks, comparable to the encoder-decoder architecture.

3.8 Text Generation using Dependency Tree Traversal

Y Park et. al. [2] comprehensively details the drawbacks of using sequence decoding. The research focuses on creating sentences with the main aim to generate text for spoken dialog systems (SDS) which acts as an important aspect of human machine interaction.

Sequence decoding though popularly used has a flaw of weak correlation between its generated words. So, the paper [2] suggests a different approach called tree dependency decoding to overcome the disadvantage of sequence decoding. In contrast to previous usage of dependency parsing, where the primary goal was to encode the sentence or analyse the sentence in a better way, the authors of the paper [2] use dependency parsing primarily for decoding during sentence generation thus primarily focusing on the ability of dependency parsing to create dependency trees.

The suggested method in the research paper involves a preprocessing phase with the use of slot chunking. Slot chunking is important to create chunks (word sequences), analysing the word sequences and giving the slot-value that matches with the best word. After that a combination of sentence analysers, dechunking and delexicalization to create input that has to be used further.

RNN requires data to be fed sequentially so, the author suggests a unique approach of converting the dependency tree to a sequence.Since the sequence of words is important for text generation in later stages, Y Park et al. works on a unique sequentialization approach, which can be reversed, by adding special markers which define the direction of motion in which tree traversal takes place. Though this traversal method does seem similar to a standard breadth-first-search approach on the tree, it is characterized by uniquely employing the use of markers. The working of this approach is shown thoroughly in Figure 6. This generated input is thus more relevant and helps to generate better sentences.

Finally, in the last phases. the authors of the paper [2] feed this sequence of inputs in the RNN to get a sequence of previously defined special markers as output which can then be used to create a dependency tree. This tree can then be easily converted back and lexicalized to a value using the slot-key as input.

The output of this unique method was that text generated seemed much more natural than text generated sequence decoding.

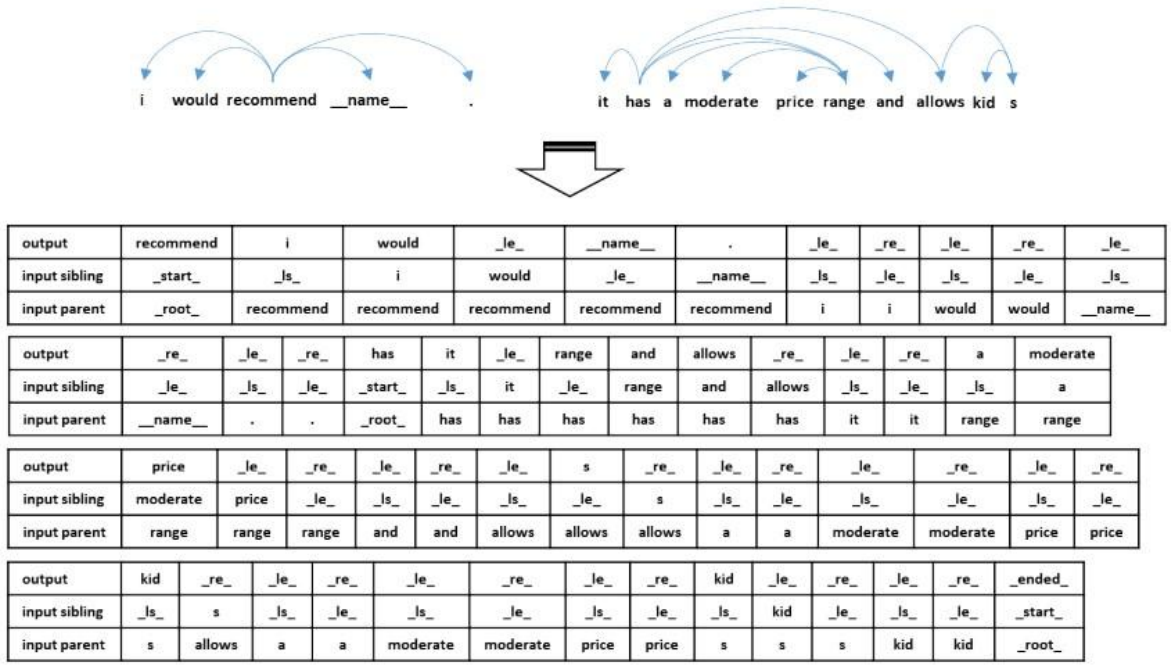

Figure 6: Example depicting how dependency trees are converted to sequence. Y Park et al. [2]

4. EVALUATION OF TEXT GENERATION MODELS

As seen in the survey so far, generating text can be difficult to work on and understand. However, how can the precision of the text generated by models be judged?Researchers are still struggling to find the most relevant metrics to evaluate the generated text.

In this section, some popular methods used to evaluate text generation models are discussed.

4.1 Human Assessment

Human assessment is a way to study and rate the quality of the generated text by a professional. As there is no proper way to find the accuracy, this way can be used depending on the use case.

As stated by Asli et al. [15], NLG tasks are open ended. A machine cannot evaluate a text generation as accurately as a human owing to numerous factors. For eg. A document being recapitulated in different ways might be correct and thus introduces a subjective nature to evaluation which cannot be accurately captured by an automatic evaluator.

4.2 Metrics

Instead of rating the quality of the text by a person, automated metrics can be used to evaluate the text. These are the most widely used and can see in most of the research papersthat have been referenced and reviewed in this survey. These metrics generally work by the comparison of two texts which are target and generated texts. Following is a brief overview of common metrics used for evaluation.

4.2.1 Bilingual Evaluation Understudy Score(BLEU)

This model calculates the n-gram overlap of the generated text and the reference corpus. It also means it is independent of the word position This is a precision focused metric and most popularly used. This metrics can be used for text summarization, machinetranslation, paraphrasing systems etc. However, there are various shortcomings for the bleu score like, it does not focus on whether all reference words are covered by the candidate. It can also not handle the semantic similarity between the words, for example “The glass is on the table”is similar to “The glass is over the table”. Bleu score cannot capture this.

4.2.2 Recall Oriented Understudy for Gisting Evaluation(ROUGE)

ROUGE is generally used with BLEU and is quite similar to it. As seen in the previous sub-topic BLEU is precision based, whereas ROUGE is recall based. There are three types of rouge:

- ROUGE-N: It is the most commonly used type and is based on n-gram overlap of reference and model generated text. Unigram would consist of one word and similarly for bigram and trigram etc.
- ROUGE-L: It computes the LCS (Longest Common Subsequence) between the output and the reference. The idea is that the greater the LCS, the more similar the phrases will be.
- ROUGE-S: It is called skip gram, as it allows us to look for words from the reference text that appear in the model output but are separated by one or more words

4.2.3 Perplexity

It is a widely used metric for evaluating the performance of generative and language models, is a measure of the likelihood that a sentence would be produced by a model trained on a dataset and is a model dependent score.

Perplexity however does not prove to be useful in cases where the diversity of the generated text has to be measured [17] and it only captures diversity to some extent. Perplexity should ideally not be used to as a metric to evaluate for GANs (which

are not language models).

4.2.4 Metric for Evaluation of Translation with Explicit Ordering(METEOR)

METEOR is another evaluation statistic that is considered to have a greater association with human assessment.This metric overcomes all the shortcomings which are experienced in BLEU.

METEOR substitutes a weighted F-score based on mapping unigrams and a penalty function for precision and recall computations for inappropriate word in order to overcome this problem. Generally, WordNet or Porter stemmer are used. Finally, an F-score is calculated using these mappings. Finally, the METEOR score is given as "(1-*Penalty*) *F".*

METEOR is a lesser-known measure in NLG, particularly after the rise of deep learning models.

5. RESEARCH GAPS

In this section, some study topics for researchers have been identified that will require significant work to increase the scope of NLG. Although it is possible to generate text using various methods, there are still several points which hamper the performance of text prediction.After reviewing various studies, most of the papers on which the researchers were working had a very limited and unstandardized source of dataset. Some work can be done to make a standardised dataset for various languages as well similar to the work done in [14].Another issue which can be highlighted from reviewing various studies is the quality assessment of text generation. In [18] the authors have also mentioned how the current quality metrics can fail to capture semantics of a text.Controllable text generation is at a very early stage.

Future research can explore multi-grained control and create suitably directional Models.PLMs are one of the best ways for natural language generation and can be used in various use cases. But still a lot of work can be done like investigating a wider variety of optimization strategies that can combine the benefits of existing approaches similar to the fine tuning of PLM.

A line of research which is being pursued in text generation called Knowledge-Enhanced Text Generation (KETG) is fairly new. While responding to any prompt (question, comment, dialogue from someone), human language is comparatively much richer and full of information, not only limited by the contents of the input to the human but also incorporating knowledge which was previously understood. This concept of not just relying on the input given to an algorithm but also using other broader sources of knowledge while giving a response is the main base for KETG NLG. Wenhao Yu et al. [5] expands on the different methods based on KETG and how it can be developed further to give much better text generation models in different scenarios.

We discussed some important evaluation metrics in Section 4.2. However, as mentioned by Iqbal and S. Qureshi[15], GAN does not have evaluation metrics to measure its accuracy properly. In most cases, N-gram overlapping evaluation metric shows very less correlation and low robustness. Currently, GAN is evaluated as standard probability-based evaluation metric thus introducing a possibility of future improvement.Stanislau et al. [16] does show that Frechet InferSent Distance (FID) and reverse LM can be used to detect shortcomings which BLEU is not sensitive to and suggests that multiple metrics need to be reported to get an accurate picture of GAN.

6. CONCLUSIONS

This paper presents an overview of the various models and recent advances achieved in text generation.Deep learning provides us a way to strap huge amounts of computation and data with small efforts done by hand.

This review explores and compares various different techniques for text generation from older techniques like Markov chains, lstm etc., to recent advancements in techniques like VAEs, GANs etc., which are very capable of generating human-like text and context aware text.The survey also discusses how pretrained language models (PLMs) use various transformers to generate better text.

If all the language models are compared, every model has their own advantages and disadvantages like:

- RNNs cannot create coherent long sequences.
- The texts which are generated by the use of LSTM are not very effective even though LSTM can generate more natural texts than Markov Chains, but it was found that Markov Chains outperformed both LSTM text generation technique and the baseline technique which is an over-sampling technique.
- Encoder-Decoder models enhanced with Attention Mechanism could perform better than RNNs but worse than Transformers.
- Transformers are novel models but they require much more data to be trained with. It can also be concluded that the area of continuous data (images) is dominated by GANs while discrete data (text) is dominated by Variational Auto-Encoders, which are the most recent advancements in the text generation field.

7. REFERENCES

[1] Akkaradamrongrat, S., Kachamas, P., & Sinthupinyo, S. (2019). Text Generation for Imbalanced Text Classification. 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE).

[2] Y. Park and S. Kang, "Natural Language Generation Using Dependency Tree Decoding for Spoken Dialog Systems," in *IEEE Access*, vol. 7, pp. 7250-7258, 2019

[3] Shuohua Zhou, (2020) " Research on the Application of Deep Learning in Text Generation" , in School of Information, Renmin University of China, BeiJing.

[4] Chenhan Yuan, Yi-chin Huang, Cheng-Hung Tsai, (2020) "Efficient text generation of user-defined topics using generative adversarial networks", in CC-NLG 2019 (Workshop on Computational Creativity in Natural Language Generation).

[5] WENHAO YU, CHENGUANG ZHU, HENG JI, (2021) "A Survey of Knowledge-Enhanced Text Generation", inarXiv:2010.04389.

[6] Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li Baobao Chang, Zhifang Sui "Order- planning Neural Text Generation From Structured Data" Sep 2017

[7] Prasad Kawthekar, Suvrat Bhooshan, (2020) "EvaluatingGenerative Models for Text Generation" Department of Computer Science Stanford University

[8] Sidi Lu, Yaoming Zhu, Yong Yu,(2018) " Neural Text

Generation: Past, Present and Beyond", arXiv preprint.

[9] Preksha Nema, Shreyas Shetty, Parag Jain, Anirban Laha, Karthik Sankaranarayanan, Mitesh M. Khapra, "Generating Descriptions from Structured Data Using a Bifocal Attention Mechanism and Gated Orthogonalization",arXiv:1804.07789v1 [cs.CL] 20 Apr 2018.

[10] Junyi Li and his team (2020) "Pretrained Language Models for Text Generation: A Survey", arXiv preprint arXiv:2105.10311

[11] Sheikh Abujar, Mahmudul Hasan, (2019) "Bengali Text generation Using Bi-directional RNN", in 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[12] Zhiqiang Ma, Chunyu Wang (2020) "A Context-Aware Variational Auto-Encoder Model for Text Generation " in IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking.

[13] Iqbal and S. Qureshi, (2020) "The survey: Text generation models in deep learning", Journal of King Saud University – Computer andInformation Sciences

[14] P. P. Barmana, A. Boruaha, "A RNN based Approach for next word prediction in Assamese Phonetic Transcription," In Procedia Computer Science, Volume 143, Pages 117- 123, 2018.

[15] Asli Celikyilmaz, Elizabeth Clark (2018) Evaluation of Text Generation: A Survey, arXiv:2006.14799v2

[16] Stanislau Semeniuta, Aliaksei Severyn (2019) On Accurate Evaluation of GANs for Language Generation arXiv:1806.04936v3

[17] Guy Tevet, Jonathan Berant (2021) Evaluating the Evaluation of Diversity in Natural Language Generation Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics.

[18] Fatima, N., Imran, A S., Kastrati, Z., Daudpota, S M., Soomro, A. (2022) A Systematic Literature Review on Text Generation Using Deep Neural Network Models IEEE Access, 10: 53490-53503