iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VII **Month of publication:** July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73254

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Automated Text Generation Applying Utilitarian n-grams

Sanjay Kumar[1], Ms. Kusum Sharma[2]
[1]*Student, Computer Science & Engineering, RSR-Rungta College of Engineering and Technology, Bhilai, CSVTU, India*
[2]*Assistant Professor, Computer Science & Engineering, RSR-Rungta College of Engineering and Technology, Bhilai, CSVTU, India*

Abstract: ***A language is a careful articulation of an artefact which can be at a basic level – a noun, an article, a verb, and adjective, a preposition, a connective, a clause, an adverb, and certain amount of punctuation. From the ancient ages, people have learned a certain amount of language for effective communication. It started with alphabets which can be connected to form a word and later leading to the output of a sentence. Lately Artificial Intelligence (AI) and Machine Learning (ML) have been developing a language model which can assist a technician in the production of a sentence. The most common techniques are Recurrent Neural Networks (RNNs), Long and Short Term Memory (LSTMs), Convolutional Neural Network (CNNs), Gated Recurrent Networks (GRUs) and others. In order that a sentence can be output the basic model of a Noun Phrase and a Verb Phrase has to be applied. In this research article we present an algorithm called ANYA (Polynomial Approximation) in order to help in the output of a sentence.***
Keywords: Anya, Artificial Intelligence, Machine Learning, Sentence Creation, LSTM.

I. INTRODUCTION

A typical sentence in the English language has syntax, and semantics. Syntax can be defined as certain rules pertaining to the structure of a sentence. Semantics can be understood as the underlying meaning formed by composition of a sentence.

A typical sentence can be defines as having the following structure

1) Noun Phrase
2) Verb Phase

A noun phrase is formed by pre-pending a article which is typically A, AN, The etc. to a noun which is typically either a name or a qualified name – like boy, tree, John etc.

Here are examples of sentences

1) A boy on a bicycle
2) John is eating
3) A tree is beautiful

The following is the Noun Phrase Verb Phrase structure of the sentences

1) A boy – Noun Phrase, on a bicycle – Verb Phrase
2) John – Noun Phrase, is eating – Verb Phrase
3) A tree – Noun Phrase, is beautiful – Verb Phrase

The following technologies can be of help in generating a sentence

1) Artificial Neural Network (ANN)
2) Long and Short Term Memory (LSTM)
3) Convolutional Neural Network (CNN)
4) Recurrent Neural Network (RNN)
5) Gated Recurrent Unit (GRU)
6) Support Vector Machine (SVM)

©IJRASET: All Rights are Reserved

The following figure shows the basic architecture of a system that can generate a sentence

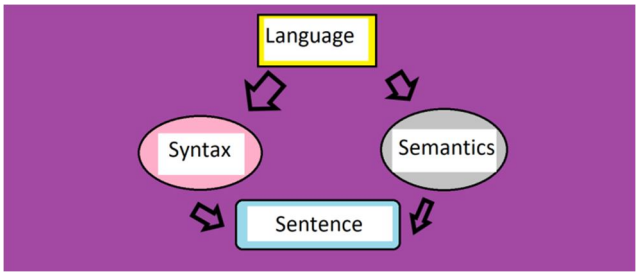


Fig.1. Sentence Generation

This research article presents a novel algorithm called ANYA (Polynomial Approximation) in order to generate simple sentences.

II. RELATED RESEARCH WORK

Here's a review of the body of work done on text generation by applying various kinds of methods.

In [1] (Daza, A., Calvo, H., Figueroa-Nazuno (2016)) look at mechanized sentence creation by learning from corpuses.

Typically, the work involving mechanized story generation is based on a general design for sentence creation.

Moreover, the results are not consistent with a literary style. The authors have a belief that in order to mechanistically create stories that could stand beside the work of human authors, a particular mechanism for fictional content should be attributed.

The authors also hold a belief that it is necessary that for a tale to carry the basic effects of the generality to the reader. The author's methodology recommends generation of stories based on corpus in order that there can be generality when it comes to tales and that there can be a fictional text.

The authors also display that these tales have general syntax and semantics and they can be considered as typical by human observers.

In [2] (Chen, S., Wang, J., Feng, X., Jiang, F., Qin, B., Lin, C. Y. (2019)) try to enhance neural data-to-text creation models with extraneous subjective knowledge.

Current neural mdoels for data-to-text creation depend on large parallel couples of data and text in order to interpret writing info. Typically, they presume that writing info can be gathered from the data that is pertaining to data. Also, when people are writing, they rely not only on data but also consider info.

In this research article, the authors enhance info with extraneous knowledge in a basic and effective way in order to better fidelity of created sentences.

Also, the authors rely on parallel data and info as in related work so that their model can standby extraneous info combining it with data in order to generate a sentence.

This can now allow the model to look at facts which are not particularly present in data from an extraneous source of info.

Empirical results on twenty-one datasets from Wikipedia show that this model is able to give better results than the state of the art.

There is a metric that is defined by the authors in order to quantify the time when the extraneous knowledge is effective.

The results of this research work display the importance of extraneous info and the basic data as the main attributes that affect the performance of the system.

In [3] (Yang, R., Zeng, Q., You, K., Qiao, Y., Huang, L., Hsieh, C. C., Rosand, B., Goldwasser, J., Dave, A., Keenan, T., Ke, Y., Hong, C., Liu, N., Chew, C., Radev, D., Lu, Z., Xu, H., Chen, Q., Li, I. (2024)) consider Ascle – a natural language processing toolkit for the generation of sentences in medical context by performing an evaluation based study.The manual management of medical info bases has always been a time consuming and a labour-intensive process.

In order to address this, algorithms pertaining to natural language processing have been developed in order to assist in sentence generation. In the bio-technology subject, different kinds of toolkits for sentence generation are present which have mostly offered their assitance in the management of unstructured info.

But, these toolkits that are present can look at different viewpoints, yet cannot offer symantics leading to a basic gap in the present offerings.

This research work intends to offer the development and basic evaluation of Ascle. Ascle is pertaining to biotech researchers and medical staff that requires a certain amount of programming expertise.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

In the present era, Ascle provides four different viewpoints

1) Fact demonstration
2) Summary of sentence
3) Simplification of sentence
4) Transliteration

Also, additionally, Ascle incorporates 12 important NLP procedures along with search functionality for medical info.
The authors have adapted 32 domain-pertaining language based models and put them through 27 standard benchmarks.
Moreover, the authors have developed a frameworks for a language model that consisted of a medical knowledge graph taking into account tanking in order to improve the performance.
Also, a validation of physicians has also been performed in order to evaluate the quality of created sentences.
The attuned models and framework lead to an improvement in sentence generation. Typically, the attenuated models lead to betterment in transliteration task by 20.3 in the BLEU score.
This study was performed on the tool Ascle which is a user-friendly NLP app for the creation of medical sentences.
In [4] (Wang, Y., Jiang, J., Zhang, M., Li, C., Liang, Y. (2023)) look at mechanized assessment of tailored sentence creation by applying language models.
Tailored sentence creation represents a particular mehcnaim for producing sentences that are particular to a user's typical context. While the scientific progress in this subject has been fast, the assessment is still a challenging task. Contemporary mechanized metrics such as ROUGE and BLEU typically evaluate the closeness of lexicons while assessing human generated references.
Also, at the same time, human evaluations can be cost pertaining, typically in the subject of customized assessment.
Getting an inspiration from these challenges, the authors explore the use of language models for assessing customized sentence creation while looking at the ability to comprehend very little context.
The authors present a novel assessment method that looks at three major aspects pertaining to semantics for generated sentences – customization, quality and pertinence.
In order to validate the usefulness of the method, the authors have developed basic empirical data while comparing the precision of assessment judgement made by language models vis a vis the assessments made by annotators.
The authors have also performed a good analyses of the nature of the metric. The authors discover that when comparing with metrics that are present, the new metric not only looks at models based on their customization but also considers the efficacy involved.
The work recommends that a language model is useful when it comes to sentence generation and is better than contemporary models while there are still a lot of issues.
In [5] (Pawade, D., Sakhapara, A., Jain, M., Jain, N., Gada, K. (2017)) consider mechanized sentence creation by applying word level RNN-LSTM.
With the invention of AI, the way that technology assists people has taken to contemporary art. From the diverse fields of music, medicine, finance, gaming, and other such domains there is now the need for extensive research and large bodies of info have to be developed for AI/ML.
A neural network is one of the many ways of AI/ML. In this research article, the authors have developed an RNN based scheme for sentence generation.
The authors discuss two possibilities when considering the nature of the input fables. At first, we have evaluated fables with varying cast and personality. Also, the authors have worked with various volumes of fables where the characters are not in harmony with each other while also offering similarities.
The results constructed by the system are then analyzed based on attributes like prosody, ethos, and prologue.
In [6] (Celikyilmaz, A., Clark, E., Gao, J. (2021)) perform an assessment of sentence creation.
The research article review assessment methodologies of natural language processing (NLP) systems that have been attributed in the recent past. The authors group NLP assessments methods into three sub-heads – (1) human based assessment rubrics (2) mechanized metrics that related to assessment (3) machine related rubrics.
For each sub-head, the authors discuss the way that has been made along with the issues that are involved while at the same time concentrating on the assessment of currently recommended NLP tasks and neural NLP models.
The authors then display two samples for job-related NLP assessment for mechanized sentence summary creation and conclude the research work by suggesting future directions.

In [7] (Henestrosa, A. L., Kimmerle, J. (2024)) comprehend the perception of mechanized sentence generation in the populace by look at two surveys with represted samples in Germany.
Mehanized sentence generation (MSG) technology has been old fashioned and it has helped in the evolution of AI. Also, it seems like tools like ChatGPT can help in the creation of a sentence. To further refine how people interact with such tools in the future, it is of paramount importance that the nature and belief of the populace be taken into consideration.
MSG research and its perception has not evolved completely. As per two attributed reviews, the authors intend to consider the concepts and beliefs of AI based sentence generation amongs the German commons.
The results revealted that there exist preference for human output on a good range of topics while there is a lack of awareness about MSG.
By utilizing a multiple model approach, the authors have looked at peoples consideration of MSG, the expectation of performance, the contemporary systems, and evaluation of tools and techniques.
The authors explore the results against the persona of mechanized context by inviting societal debate about the offerings involved.
In [8] (Karkouri, A. A., Lazrak, M., Ghanimi, F., Amrani, H. E., Benammi,D., Bourekkadi, S. (2023)) consider the creation of mechanized texts and reports for the subject of contemporary analysis by examining deep learning.
This research article recommend a novel methodology applying deep learning to mechanize the creation of sentences in reports that explain basic economics.
It also looks the effect on various sectors of public utility. Decision making depends on the info, firms outlook, consumers, the government and a lot of other factors.
Report writing has been considered as part of this research work. The research work then displays deep learning as a good offering for looking at info.
The “Commodities and Processes” part furnishes basic explanation of the info, pre-processing of the data, development of the model, training and evaluation.
These basic and necessary mechanisms are necessary to provide info. The research essay brings to light how deep learning scheme may increment the accuracy of economic arena while at the same time considering other factors. The goodness of this framework can be assessed by looking at fashioned instance of report generation.
This research article documents how mechanization is now banking on contemporary methods when considering economic research. It offers a bird’s eye view of economic research, thus displaying that deep learning can help and assist in fashioning info.
This growth has inculcated a good novel resource for economists so that decision making can involve the utliziation of large amount of info.
In [9] (Kumar, M., Kumar, A., Singh, A., Kumar, A. (2021)) consider the evaluation of mechanized sentence creation by applying deep learning. A chatbot is a computer paragraph that can talk to humans by applying AI while tending towards messaging systems.
The objective of this research work is to utilize and improve deep learning techniques thus building a good chat bot.
In the present era of chat bots various developments have progressed applying rule based methods, basic ML procedures or fetch based schemes which can lead to dialogue.
The authors have contrasted the performance of RNNs, GRUs and LSTMs. The automation that can be derived can now create dialogue.
In [10] (Harrison, B., Purdy, C., Riedl, M. O. (2017)) lean on the sentence creation with HMMs and Monte Carlo methods by applying deep learning.
In this research article, the authors introduce a method for mechanized sentence creation by applying HMMs and Monte Carlo methods. This scheme applies a consideration based approach that utilizes a Metropolis-Hastings to create a proabibilistic distribution which can then be applied to create fables.
The authors display the fact that the application of various schemes has to be taken into account for the creation of sentences and the info can be taken from movie clips and from the song sequences.
This research article also shows that sentences that are output from this project are quite basic in nature and can be following preliminary criterion of assessment.
In [11] (Hervas, R., Pereira, F. C., Gervas, P., Cardoso) present a simplified sentence generation. In order that simple info be communicated there has to be the correct usage of analogy in computer generated sentence.
This research paper intends to better the fashioned attributes of sentences generated by a natural language processing system by taking the help of an analogy.
A simple architecture and a particular process for example can be addressed.

This demonstration intends to form a multi-example design for a particular implemtation of a module that are interlinked while looking at a sub-procedure of a process – thus enhancing domain based info, discussing subject involvement, and appending examples in the generation of a sentence.

Basic results are discussed, and various issues resulting from these behaviour are assessed paying little attention to the related betterment of the proposed design.

In [12] (Upadhyay, L., Hasan. M. I., Patel, P. S. (2023) look at simple sentence generation approaches. NLP is a subject of AI that concentrates on making machines look at language and interact with people.

The primary focus on the job of sentence creation is to create certain type of sentences. At a certain level the technique has involved training a neural network model that consists of an encoder that can generate a representative sentence along with a decoder model to have various interpretations of the sentence.

For the job of sentence generation, different processes and models can be chosen. In the sub-sections of this research article various procedures as per the current project of generating a sentence are discussed.

In the subject of sentence generation, the researchers primary goal is to utilize an HMM and LSTM units in order to create a sentence.

In [13] (Layne, S., Gehrmann, S., Dernoncourt, F., Wang, L., Bui, T., Chang, W. (2022)) look at a conceptualization for mechanized sentence generation.

Researchers in subjects such as transliteration and summary creation have to analyse results based on a broad range of publishing guidelines that typically apply different assessment schemes.

The authors intend to safeguard a simple comparison by displaying certain benchmarks on a tool for looking at the creation of a sentence.

Sentence generation processes and assessment rubrics can simply be augmented to a benchmark and its results can be looked at by analysing the varieties of methods in which users supply a large number of corpora, systems, and assessment methodologies that can then fetch comparison reports in a graphical and tabular format.

In [14] (Iqbal, T., Qureshi, S. (2020)) perform a review on various sentence generation models that apply deep learning.

Deep learning procedures require a lot of layers to comprehend the structured representation of info and have led to the state of the art in some fields.

Currently, Deep Learning Models and projects have unveiled in the subject of Natural Language Processing (NLP). This review furnishes a little description of the progress that has hapenned in the subject of Deep Learning Modelling.

This research work takes into account some of the research article from 2015 and onwards. In this research project, the authors have reviewd many Deep Learning Models that have been presented for the generation of a sentence.

The authors have also presented a summary of a certain number of models that have been put forth in the past for sentence generration applying deep learning.

Moreover, a certain amount of Deep Learning methods have been examined and assessed in various info subjects concerning NLP and they have been included in this review.

In [15] (Upadhyay, A., Massie, S., Singh, R. K., Gupta, G., Ojha, M. (2021)) present an approach that involves info to sentence generation.

Contemporary info-to-sentence generation (I2S) processes apply subject specific instances and templates in order to produce simple sentences. Very recent methdologies apply neural systems to grasp subject info from structured data to generate basic sentence output.

Moreover, there is a need to strike a balance between rule-based schemes that can generate output but these may lack variety while analogy based systems might be producing mixed output with little accuracy.

In this research article, the authors recommend Info to Sentence (I2S) that reduces the impact of these trade-offs by selecting corpora.

In this scheme, the authors construct a new subject based attributed method that can be utilized to construct a similarity measure that is applicable in certain cases.

Good empirical processes can be looked at in certain subject areas. By applying certain assessment based rubrics, the authors display the benefits of the I2S system over a rule-based benchmark.

In [16] (Gayam, S. R. (2022)) explain generative AI for creation of a sentence by looking at modern methodologies for mechanized sentence generation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The new field of Artificial Intelligence (AI) has now turned towards novel subjects like generative models, and is quite capable of creating basically a simple sentence.
The research article looks at the research involved while examining the strengths and the limitations.
The research article begins by looking at the arena of natural language processing (NLP). The sub-areas that are exmained are RNNs, LSTMs, and GRUs.
The discussion looks at transformers, that have helped out RNNs. Later the research article delves into computer vision while examining GANs.
Various type of GAN architectures are explored which include Deep Convolutional GANs and other variants like StyleGANs which can demonstrate a simple sentence.
This research article intends to furnish a thorough review of a certain number of techniques in AI that can be then utilized for the creation of a simple sentence.
In [17] (Li, J., Tang, T., Zhao, W. X., Wen, J, R. (2021)) consider trained models that can do simple sentence generation by performing a review.
Sentence generation involves the application of Natural Language Processing. The application of deep learning has also advanced this field by looking at trained models. In this research article the authors present a review of the leaps made in the topic of NLP.
As per the preliminaries, there is the progression of a definition and then certain paradigms of sentence generation are discussed. The authors later present a summery for various pertinent strategems for the creation of a sentence. Also, the authors then present directions in the future as they conclude the research article.
In [18] (Guo, Q., Qiu, X., Xue, X., Zhang, Z. (2019)) look at sentence generation with syntax by applying an ANN. Sentence generation is a basic task in N:LP. A large number of existing models generate basic sentences and require modelling.
In this research article the authors treat the sentence generation job as a graph theoretical problem involving syntactic sugar.
As per the process, the ANN builds a sentence that is simple while maintaining syntax in a top down and breadth first manner.
Empirical reults on the real world and syntactic generation shows a basic model.
In [19] (Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E. P. (2017)) examine the generation of a sentence. Basic generation and sentence modelling is a challenging task which requires some deep generative analysis in a graphical manner.
This research article intends to generate a simple sentence whose traits are rendered by a learning latent disentagled representation of semantics.
The authors recommend a novel generative model which can look at variational auto-encoders (VAEs) while thinking about a guided discrimination for examination of semantic structrures. Alternatively, the models can be seen as enhanced VAEs with the algorithm that can produce a sentence. With a discriminatory guess of a discrete sentence, and constraints on attributed data the model can render a basic sentence.
Qualitative analysis applying trained classifiers as assessors can help output a sentence.

III. METHODOLOGY

The following is the methodology of the ANYA algorithm

1) Input Data
   a. Three data files data-1.txt, data-2.txt and data-3.txt are input and contain 3 template sentences each
   b. A thesaurus – thesaurus.txt containing word to number matching is entered
   c. An info – info.txt file containing correctness is input

2) Convert Data
   a. The input sentences from data-1.txt are converted to polynomials
   b. The thesaurus is loaded and kept as dictionaries
   c. The info file is loaded and kept as lists inside dictionaries

3) Generate Random Sentence
   a. A random sentence is generated by taking random samples from the polynomials

4) Output Right Sentence
   a. A correct sentence is output by approximating a correct polynomial taking help from the info correctness.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The following are the details of the steps in the methodology
Input Data:
The Input Data has been carefully constructed so that there is just enough variety and structure to the sentences.
The following are the files
data1.txt
Joe might need to sleep
Joe is going to playground
Joe can be at club
data2.txt
John has slept till late
John will need a soda
John is looking at food
data3.txt
Jack may go to cinema
Jack is eating at stall
Jack can try out clothes
The following is a snippet of the file thesaurus.txt
Joe, 1
John, 2
Jack, 3
might, 11
is, 12
can, 13
has, 14
will, 15
may, 16
need, 21
going, 22

The general idea is that the proper noun phrase (1st word of every sentence) is from 1 to 3. Joe is 1, John is 2, Jack is 3.
The connectives (is, can, has, will, may – 2nd word of every sentences) is from 12 to 16.
And so on and so forth.
The thesaurus is for creating a polynomial from the data1.txt, data2.txt and data3.txt. The general idea being to have a number based representation of the sentences.
The info file is the correct possible interpretations of all the possible mixtures of the sentences and is as follows
Joe, may, be, at, sleep
Joe, will, need, to, sleep
#
Joe, can, be, at, playground
Joe, is, looking, at, playground
Joe, might, try, out, playground
#
Joe, may, try, out, club
Joe, will, go, to, club
Joe, has, slept, at, club
#
Joe, can, be, till, late
Joe, is, going, at, late
Joe, might, be, till, late
#

Joe, may, need, a, soda
Joe, will, try, a, soda
#
Joe, can, try, out, food
Joe, is, going, to, food
#
Joe, may, try, out, cinema
Joe, will, be, at, cinema
#
Joe, can, go, to, stall
Joe, is, looking, at, stall
Joe, might, try, out, stall
#
Joe, can, try, out, clothes
#

Convert Data:
At the beginning of the conversion the input sentences are converted to polynomials. For instance the sentence
Jack can try out clothes

Is expressed as follows
'factor_num': 1,
'pos_tag_num': <PartsOfSpeech.PROPERNOUNPHRASE: 3>, 'val': '3

What this means is that the ProperNounPhrase (factor number 1) has the value 3 (Jack is 3 in thesaurus.txt)
'factor_num':2,
'pos_tag_num': <PartsOfSpeech.CONNECTIVE: 4>,
'val': '13',

What this means is that the Connective (factor number 2) has the value 13 (can is 13 in thesaurus.txt)
'factor_num':3,
'pos_tag_num': <PartsOfSpeech.VERBPHRASE: 5>,
'val': '29',

'factor_num':4,
'pos_tag_num': <PartsOfSpeech.PREPOSITION: 6>,
'val': '35',

'factor_num':5,
'pos_tag_num': <PartsOfSpeech.NOUNPHRASE: 7>, 'val': '49'

The thesaurus is loaded and is kept as a dictionary
key: Joe val: 1
key: John val: 2
key: Jack val: 3
key: might val: 11
key: is val: 12
key: can val: 13
key: has val: 14
key: will val: 15

key: may val: 16
key: need val: 25
key: going val: 22
key: be val: 23
key: slept val: 24
key: looking val: 26
key: go val: 27
key: eating val: 28
key: try val: 29

The info file is loaded and is kept as a list (where the key is the NounPhrase) inside a dictionary
For instance the info sentences
Joe, may, be, at, sleep
Joe, will, need, to, sleep

Are kept as

key: sleep val: [['16', '23', '32'], ['15', '25', '31']]

where,

16 = may
23 = be
32 = at

The info sentences

Joe, can, be, at, playground
Joe, is, looking, at, playground
Joe, might, try, out, playground

Are kept as

key: playground val: [['13', '23', '32'], ['12', '26', '32'], ['11', '29', '35']]

where:

13 = can
23 = be
32 = at

Generate Random Sentence:
A random sentence is generated by taking random samples from the polynomials.
For instance a random sentence example is:
anya: got curr_new_sentence Jack may eating to late
Output Right Sentence:
A correct sentence is then output by matching with the info lists inside dictionaries
anya: right_sentence Jack might be till late

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

IV. RESULTS

At first a thesaurus is loaded and here's a snippet of output of anya
read_thesarus: Joe, 1
read_thesarus: John, 2
read_thesarus: Jack, 3

An info is loaded and here's a snippet of the output
read_info: Joe, may, be, at, sleep
read_info: Joe, will, need, to, sleep
read_info: Joe, can, be, at, playground
read_info: Joe, is, looking, at, playground
read_info: Joe, might, try, out, playground
read_info: Joe, may, try, out, club

The sentences are input from data-1.txt, data-2.txt and data-3.txt and polynomials are computed. Here's a snippet of the output

DataPolynomial1
curr_sentence: 1
Joe might need to sleep
[1, 1, {'factor_num': 1, 'pos_tag_num': <PartsOfSpeech.PROPERNOUNPHRASE: 3>, 'val': '1'}, {'factor_num': 2, 'pos_tag_num': <PartsOfSpeech.CONNECTIVE: 4>, 'val': '11'}, {'factor_num': 3, 'pos_tag_num': <PartsOfSpeech.VERBPHRASE: 5>, 'val': '25'}, {'factor_num': 4, 'pos_tag_num': <PartsOfSpeech.PREPOSITION: 6>, 'val': '31'}, {'factor_num': 5, 'pos_tag_num': <PartsOfSpeech.NOUNPHRASE: 7>, 'val': '41'}]

A shuffled sentence is then output. Here's a snippet

anya: curr_polynomial_list_not_shuffled [(1, 1), (3, 1), (3, 2), (3, 3), (2, 1)]
anya: curr_polynomial_list [(1, 1), (3, 1), (2, 1), (3, 3), (3, 2)]
anya: got curr_new_sentence Joe may slept out stall
anya: get_right_sentence Joe may slept out stall
anya: get_right_sentence proper_noun_phrase Joe
anya: get_right_sentence connective may
anya: get_right_sentence verb_phrase slept
anya: get_right_sentence preposition out
anya: get_right_sentence noun_phrase stall
anya: get_right_sentence 16 24 35
anya: curr_info_dict_list [['13', '27', '31'], ['12', '26', '32'], ['11', '29', '35']]

Finally a right sentence is output by anya

checking list_num: 0 [16, 24, 35]
checking list_num: 0 [13, 27, 31]
checking list_num: curr_diff 10
checking list_num: 1 [16, 24, 35]
checking list_num: 1 [12, 26, 32]
checking list_num: curr_diff 9
checking list_num: 2 [16, 24, 35]
checking list_num: 2 [11, 29, 35]
checking list_num: curr_diff 10
anya: right_sentence Joe might try out stall

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. COMPARATIVE ANALYSIS

The following are the names of two programs that the authors would like to compare Anya with

- Sentence-generator
- Basic-sentence-generator-using-ngram

The sentence-generator program has output in a following way

```
acy\sentence-generator-main>python sentenceGenerator.py
Select words: i am
Sorry! That isn't a word selection. Try one of these:
       STARGATE
       KERBAL SPACE PROGRAM
       Enter for last words
       END to end
Select words: STARGATE
Gorge Hammond carelessly searched the cool locals under a P-90.
Select words: KERBAL SPACE PROGRAM
Bill Kerman orbited around Duna, and Valentina Kerman forcefully engineered a unsafe spaceplane when Bill Kerman threw i
nto space Kerbin under the freezing space plane hanger before Jebediah Kerman blew up Duna on floating Kerbal Space Prog
ram, so Jebediah Kerman engineered the unsafe space plane hanger, but Bob Kerman engineered rusty Kerbin, so Valentina K
erman blew up the crazy space plane hanger inside of Duna, but Valentina Kerman piloted crazy Kerbin when Bill Kerman bl
ew up a noninspected spaceplane.
Select words:
Valentina Kerman orbited around the space plane hanger.
Select words:
Bob Kerman destroyed unsafe Kerbin.
Select words:
Bill Kerman destroyed Kerbal Space Program, so Bob Kerman set fire to floating Kerbin.
Select words:
Bill Kerman forcefully orbited around Kerbin on Kerbal Space Program.
Select words:
Bob Kerman experimented on a crazy building after Bill Kerman crashed a noninspected spaceplane.
Select words:
```

Fig 2: sentence-generator output

The basic-sentence-generator-using-ngram program has output in a following way:

```
In [1]: import pandas as pd
        data=pd.read_csv('Context.csv')
        data.head()

        C:\Users\Acer\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas
        \core\computation\expressions.py:21: UserWarning: Pandas requires version '2.8.
        4' or newer of 'numexpr' (version '2.8.3' currently installed).
          from pandas.core.computation.check import NUMEXPR_INSTALLED
```

Out[1]:

	Text	Context/Topic
0	The eternal mystique of Goldman Sachs	Politics
1	Either you don't care enough to actually tell ...	Love
2	I am such an IDIOT.	Heavy Emotion
3	While lifting weights on Friday and doing bent...	Health
4	Something's watching me	Animals

Fig 3: Basic-sentence-generator-using-ngram output

```
In [2]: data=data.iloc[:3000,:1]

In [3]: data
```

Out[3]:

	Text
0	The eternal mystique of Goldman Sachs
1	Either you don't care enough to actually tell ...
2	I am such an IDIOT.
3	While lifting weights on Friday and doing bent...
4	Something's watching me
...	...
2995	Internal Relationships and How they Hinder the...
2996	Hemp seed extract acted on U-87 cells by induc...
2997	What i wanna say is: Hitler is the same good a...
2998	I built this app myself for fun, because I thi...
2999	Please dm me when it's out

Fig 4: Basic-sentence-generator-using n-gram output

TEXT PREPROCESSING

Convert text to lower case

```
In [4]: data['clean_text'] = data['Text'].str.lower()
        data.head()
```

Out[4]:

	Text	clean_text
0	The eternal mystique of Goldman Sachs	the eternal mystique of goldman sachs
1	Either you don't care enough to actually tell ...	either you don't care enough to actually tell ...
2	I am such an IDIOT.	i am such an idiot.
3	While lifting weights on Friday and doing bent...	while lifting weights on friday and doing bent...
4	Something's watching me	something's watching me

Fig 5: Basic-sentence-generator-using-ngram output – 3

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Tokenization

```
In [5]: from nltk.tokenize import word_tokenize
        data['clean_text'] = data['clean_text'].apply(word_tokenize)

In [6]: data.head()
```

Out[6]:

	Text	clean_text
0	The eternal mystique of Goldman Sachs	[the, eternal, mystique, of, goldman, sachs]
1	Either you don't care enough to actually tell ...	[either, you, do, n't, care, enough, to, actua...
2	I am such an IDIOT.	[i, am, such, an, idiot, .]
3	While lifting weights on Friday and doing bent...	[while, lifting, weights, on, friday, and, doi...
4	Something's watching me	[something, 's, watching, me]

<- Performing stop word removal will be a hinderance to context for sentence generation ->

Fig 6: Basic-sentence-generator-using-ngram output

Creating the ngram model (bigram)

```
In [7]: from nltk import ngrams
        from collections import defaultdict
        def generate_ngram(words, ngrams, n):
            # Create n-grams
            for i in range(len(words) - n):
                context = tuple(words[i:i+n])
                next_word = words[i+n]
                ngrams[context].append(next_word)

In [8]: master_ngram=defaultdict(list)
        for i in data['clean_text']:
            generate_ngram(i, master_ngram, 2)
```

<- master_ngram is going to store the ngram for all the sentences in the dataset ->

```
In [9]: print(master_ngram)

defaultdict(<class 'list'>, {('the', 'eternal'): ['mystique'], ('eternal', 'm
ystique'): ['of'], ('mystique', 'of'): ['goldman'], ('of', 'goldman'): ['sach
s'], ('either', 'you'): ['do'], ('you', 'do'): ["n't", 'things', "n't", 'anyt
```

Fig 7: Basic-sentence-generator-using-ngram output

Function for Generating sentences

```
In [10]: import random
         def generate_sentence(model, sentence, n, max_length=50):
             while len(sentence) < max_length:
                 context = tuple(sentence[-n:])
                 if context in model:
                     next_word = random.choice(model[context])
                     sentence.append(next_word)
                 else:
                     break

             return ' '.join(sentence)
```

Giving input sentences

```
In [11]: input_sentence1="I want to understand"
         #preprocessing
         input_tokenized1=word_tokenize(input_sentence1.lower())
```

Fig 8: Basic-sentence-generator-using-ngram output

```
In [68]: input_sentence1="I want to understand"
         #preprocessing
         input_tokenized1=word_tokenize(input_sentence1.lower())

In [69]: # Creating ngram for the input sentence and appending it to the master_ngram
         generate_ngram(input_tokenized1,master_ngram,2)

In [70]: # Generate sentences with the input sentence
         generated_sentence1=generate_sentence(master_ngram,input_tokenized1,2)

In [71]: generated_sentence1

Out[71]: 'i want to understand how they form ?'
```

Fig 9: Basic-sentence-generator-using-ngram output

The sentence-generator program has the following steps in the methodology

1) There is a dictionary of words
2) There is a dictionary of conjunctions
3) There is a dictionary of prepositions
4) There is a function called generateSentence
    a. The function takes an argument which is words selected by a user in upper case
5) The function generateSentence works in the following manner
    a. The generateSentence function just adds an appropriate part of speech in a random manner in order to generate a random sentence

The Basic-sentence-generator-using-ngram program has the following steps in the methodology

1) There is a file called Context.csv
    a. It contains the following content
        i. Text
        ii. Context/Topic
    b. For instance here are examples of text and context/Topic
        i. The patient replied, “I, sir, am Napoleon”, Joke
        ii. He had to go back after visiting, Joke
2) The text from Context.csv is loaded into data structure called data using pandas
3) The variable data is preprocessed
    a. Its cleaned by converting it all to lower case

4) An n-gram called a bi-gram is then created from the words in the variable data
5) There is then a function called generate_sentence
6) The function generate_sentence is called with the bi-gram as a model and an input sentence
    a. The input sentence is for example
        i. I want to understand
7) The new generated sentence is generated by appending words from the model to the input sentence as per a relevant context

The sentence-generator program is able to output certain kinds of sentences by applying randomization to an input corpus and by examining the parts of speech

The Basic-sentence-generator-using-ngram is able to output sentences by looking at an n-gram formed from a context. Also an input sentence is then appended with certain words taken from an n-gram

The ANYA (Polynomial approximations) algorithm has a lot of analysis as compared to the earlier approaches of sentence-generator and Basic-sentence-generator-using-ngram.

The ANYA algorithm has text, thesaurus and info as compared to just a corpus in sentence-generator and context.csv in Basic-sentence-generator-using-ngram.

Also, the n-grams in ANYA are utiliatarian n-grams that are generated by evaluating certain corpora as polynomials.

Later the polynomials are combined in a random manner to generate a sentence which is later corrected by utilizing info.

The following bar graph, line chart and table give a pictorial comparative analysis of the programs

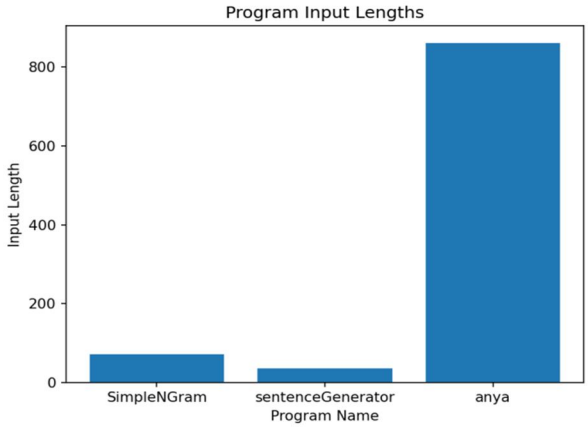


Fig 10: Bar graph of comparison

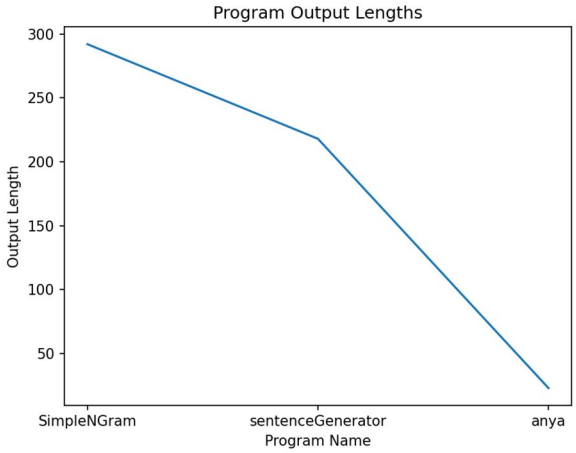


Fig 10.1: Line chart of comparison

Table 1. Running time comparison

S. No.	Program Name	Running Time
1	SimpleNGram	2.200614929
2	sentenceGenerator	0.002013206
3	anya	0.002002954

VI. CONCLUSION

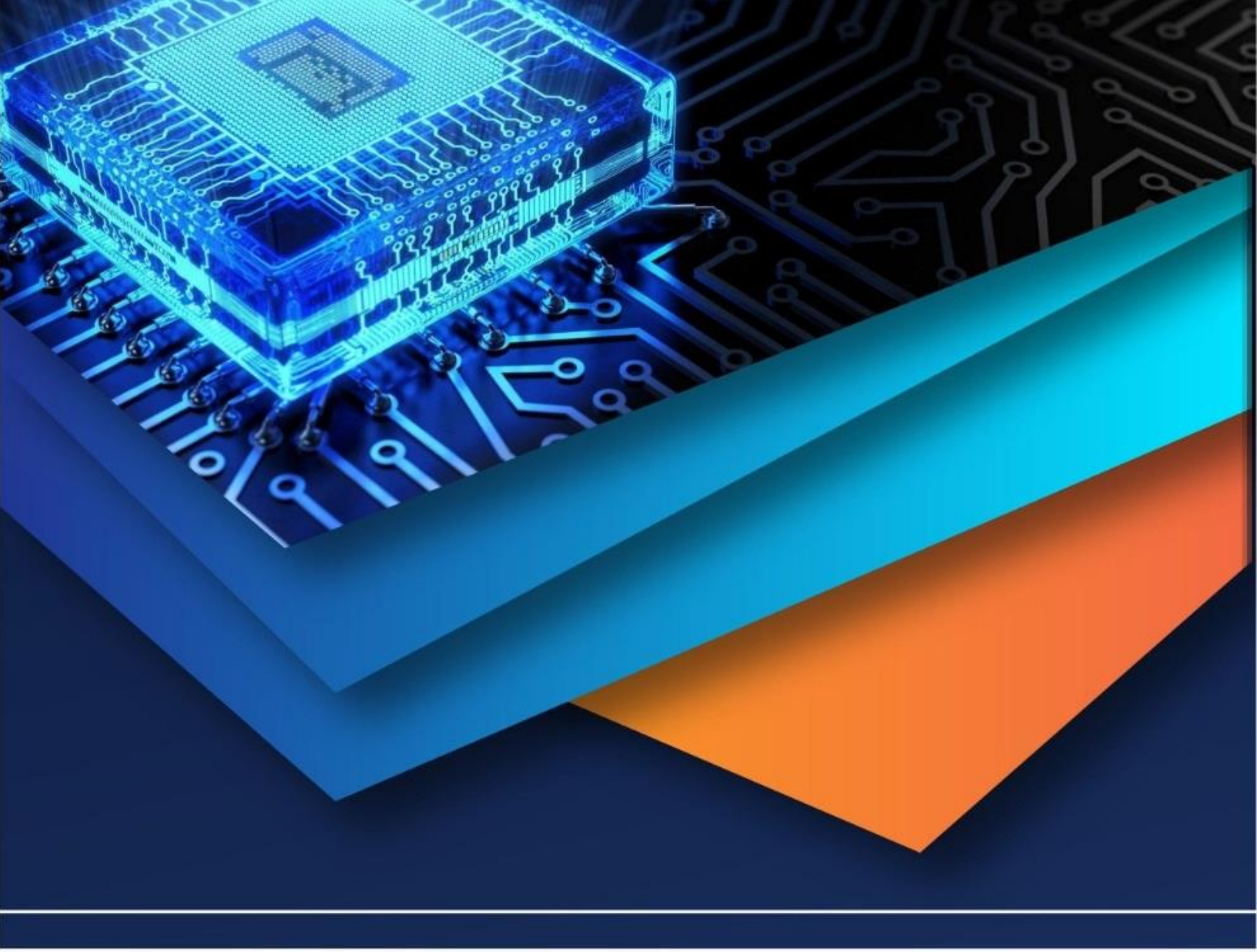
Sentence output from a computer has been studied lately. There has been active research in this area over the past decade.
The research work so far has involved various machine learning and AI based algorithms. Some of the algorithms are Deep Neural Networks, Case Based Reasoning, Neural Methods, Large Language Models, etc.
The literature review for this research article has been taken from various sources available on the internet as per certain criterion.
An algorithm called ANYA (Polynomial Approximation) is presented as part of this research. The algorithm tries to make a polynomial approximation of a sentence so that a sentence can be represented as a lit of number.
Later, the list of numbers is chosen at random in order to generate a sentence that is not exactly appropriate semantically, however it is able to follow the syntactic rules of the grammar.

The ANYA algorithm is able to generate a right sentence by looking at the correct interpretations of the sentence.

REFERENCES

[1] Daza, A., Calvo, H., Figueroa-Nazuno (2016). Automatic Text Generation by Learning from Literary Structures, Proceedings of the Fifth Workshop on Computational Linguistics for Literature.
[2] Chen, S., Wang, J., Feng, X., Jiang, F., Qin, B., Lin, C. Y. (2019). Enhancing Neural Data-To-Text Generation Models with External Background Knowledge, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3022-3032
[3] Yang, R., Zeng, Q., You, K., Qiao, Y., Huang, L., Hsieh, C. C., Rosand, B., Goldwasser, J., Dave, A., Keenan, T., Ke, Y., Hong, C., Liu, N., Chew, C., Radev, D., Lu, Z., Xu, H., Chen, Q., Li, I. (2024). Ascle- A Python Language Processing Toolkit for Medical Text Generation: Development and Evaluation Study, Journal of Medical Internet Research, Vol. 26.
[4] Wang, Y., Jiang, J., Zhang, M., Li, C., Liang, Y. (2023). Automated Evaluation of Personalized Text Generation using Large Language Models
[5] Pawade, D., Sakhapara, A., Jain, M., Jain, N., Gada, K. (2017). Story Scrambler – Automated Text Generation using Word Level RNN-LSTM, I. J. Information Technology and Computer Science, 6, 44-53.
[6] Celikyilmaz, A., Clark, E., Gao, J. (2021). Evaluation of Text Generation: A Survey, arXiv
[7] Henestrosa, A. L., Kimmerle, J. (2024). Understanding and Perception of Automated Text Generation among the Public: Two Surveys with Representative Samples in Germany, Behavioral Sciences, Behav. Sci., 14, 353.
[8] Karkouri, A. A., Lazrak, M., Ghanimi, F., Amrani, H. E., Benammi,D., Bourekkadi, S. (2023). Journal of Theoretical and Applied Information Technology, Vol. 101, No. 23.
[9] Kumar, M., Kumar, A., Singh, A., Kumar, A. (2021). Analysis of Automated Text Generation Using Deep Learning, International Journal for Research in Advanced Computer Science and Engineering, Vol. 7, Issue 4.
[10] Harrison, B., Purdy, C., Riedl, M. O. (2017). Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks.
[11] Hervas, R., Pereira, F. C., Gervas, P., Cardoso, A. Cross-Domain Analogy in Automated Text Generation.
[12] Upadhyay, L., Hasan. M. I., Patel, P. S. (2023). Demystifying Text Generation Approaches.
[13] Layne, S., Gehrmann, S., Dernoncourt, F., Wang, L., Bui, T., Chang, W. (2022). A Framework for Automated Text Generation Benchmarking.
[14] Iqbal, T., Qureshi, S. (2020). The Survey: Text Generation Models in Deep Learning. Journal of King Saud University – Computer and Information Sciences.
[15] Upadhyay, A., Massie, S., Singh, R. K., Gupta, G., Ojha, M. (2021). A case-based approach to data-to-text generation.
[16] Gayam, S. R. (2022). Generative AI for Content Creation: Advanced Techniques for Automated Text Generation, Image Synthesis, and Video Production, Journal of Science & Technology, Vol. 3, Issue 1.
[17] Li, J., Tang, T., Zhao, W. X., Wen, J, R. (2021). Pretrained Language Models for Text Generation: A Survey, Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-21).
[18] Guo, Q., Qiu, X., Xue, X., Zhang, Z. (2019). Syntax-guided text generation via graph neural network, Science China, Information Sciences, Vol. 64.
[19] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E. P. (2017). Toward Controlled Generation of Text.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)