FICTIONAL STORY GENERATOR

SANATH KUMAR NEDUNURI

This dissertation was submitted in part fulfilment of requirements for the degree of MSc Information and Library Studies

DEPT. OF COMPUTER AND INFORMATION SCIENCES

UNIVERSITY OF STRATHCLYDE

JANUARY 2022

**DECLARATION**

This dissertation is submitted in part fulfilment of the requirements for the degree of MSc of the University of Strathclyde.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the Departmental Ethics Committee as appropriate to my research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, Department of Computer and Information

Sciences, to place a copy of the dissertation in a publicly available archive. (please tick) Yes [ ] No [ \* ].

I declare that the word count for this dissertation (excluding title page, declaration, abstract, acknowledgements, table of contents, list of illustrations, references and appendices is 11408.

I confirm that I wish this to be assessed as a Type 1 2 **3** 4 5 .

Dissertation (please circle)

Signature: SANATH KUMAR NEDUNURI

Date: 31/08/2022

Table of Contents

[**Abstract** 6](#_Toc57480)

[**Chapter 1; Introduction** 7](#_Toc57481)

[**1.1 System Aims** 7](#_Toc57482)

[**1.2 System Objectives** 8](#_Toc57483)

[**1.3 System Background** 8](#_Toc57484)

[**1.4 Problem Statements** 9](#_Toc57485)

[**1.5 System Scope** 9](#_Toc57486)

[**1.6 Research Questions** 9](#_Toc57487)

[**1.7 System Functionality** 10](#_Toc57488)

[**1.8** **Operating Environment** 10](#_Toc57489)

[**1.9 Appropriated Technologies** 10](#_Toc57490)

[**Chapter 2; Overall Description** 11](#_Toc57491)

[**2.1 Existing Systems** 11](#_Toc57492)

[**2.2 Other Systems** 12](#_Toc57493)

[**2.2.1 Novel Writer System** 12](#_Toc57494)

[**2.2.2 TALESPIN** 13](#_Toc57495)

[**2.2.3 Dehn’s AUTHOR** 13](#_Toc57496)

[**2.2.4 Leobowitz’s UNIVERSE** 14](#_Toc57497)

[**2.2.5 MINSTREL** 14](#_Toc57498)

[**2.2.6 MEXICA** 15](#_Toc57499)

[**2.2.7 BRUTUS** 15](#_Toc57500)

[**2.2.8 FABULIST** 16](#_Toc57501)

[**2.3 Explications** 16](#_Toc57502)

[**Chapter 3; Literature review** 18](#_Toc57503)

[**3.1 Literature Background** 19](#_Toc57504)

[**3.1.1 GPT 3** 19](#_Toc57505)

[**3.1.2 Long Text Progressive Generation** 19](#_Toc57506)

[**3.1.3 Controllable Text Generation** 19](#_Toc57507)

[**3.1.4 Plug and Play Language Model** 20](#_Toc57508)

[**Chapter 4; Software Requirements Analysis** 21](#_Toc57509)

[**4.1 Performance Requirements** 21](#_Toc57510)

[**4.2 Safety and Security Requirements** 21](#_Toc57511)

[**4.3 Software Quality Attribute** 21](#_Toc57512)

[**4.3.1 Extensibility** 21](#_Toc57513)

[**4.3.2 Portability** 21](#_Toc57514)

[**4.3.3 Maintainability** 22](#_Toc57515)

[**4.3.4 Security** 22](#_Toc57516)

[**4.3.5 Usability** 22](#_Toc57517)

[**4.3.6 Reliability** 22](#_Toc57518)

[**4.4 Quality Assurance** 22](#_Toc57519)

[**4.4.1 Introduction** 22](#_Toc57520)

[**Chapter 5; Methodology** 24](#_Toc57521)

[**5.1 Engineering Research Methodology** 24](#_Toc57522)

[**5.2 Model View Controller Architecture** 24](#_Toc57523)

[**5.3 Methodology Investigation** 24](#_Toc57524)

[**5.4 Data Driven Story Generator** 25](#_Toc57525)

[**5.5 Research Methodology Plan** 26](#_Toc57526)

[**5.6 Database Design** 26](#_Toc57527)

[**5.6.1 Introduction** 26](#_Toc57528)

[**5.6.2 DBMS** 27](#_Toc57529)

[**5.6.3 Approaches** 27](#_Toc57530)

[**5.6.4 DB Design** 28](#_Toc57531)

[**5.6.5 Database Structure** 28](#_Toc57532)

[**5.7.5 Normalization** 28](#_Toc57533)

[**Chapter 6; Approach** 29](#_Toc57534)

[**6.1 Computational Approach** 29](#_Toc57535)

[**6.1.1 Steps Guidance** 29](#_Toc57536)

[**6.1.2 Reader Model** 30](#_Toc57537)

[**6.2 System Core** 30](#_Toc57538)

[**6.3 Non-Attractive Story Generation** 31](#_Toc57539)

[**6.4 Modelling Readers** 31](#_Toc57540)

[**6.4.1 Theory of Mind** 31](#_Toc57541)

[**6.4.2 Model Presentation and Approach** 32](#_Toc57542)

[**6.5 Composite View Point** 33](#_Toc57543)

[**6.6 Development Plan** 34](#_Toc57544)

[**Chapter 7; Gantt chart** 35](#_Toc57545)

[**Chapter 8; Analysis** 36](#_Toc57546)

[**8.1 Algorithm Analysis** 36](#_Toc57547)

[**8.2 Solution 1** 37](#_Toc57548)

[**8.3 Solution 2** 37](#_Toc57549)

[**8.4 Other System Solution** 37](#_Toc57550)

[**8.5 Measured and Analyze Solutions** 38](#_Toc57551)

[**Chapter 9; User Manual** 39](#_Toc57552)

[**Open Main.py in visual Studio** 39](#_Toc57553)

[**MVC Architectural Model** 39](#_Toc57554)

[**Index.html** 40](#_Toc57555)

[**Installing Packages** 40](#_Toc57556)

[**Application Loading Dashboard** 41](#_Toc57557)

[**Application Dashboard** 41](#_Toc57558)

[**Entering Attributes** 42](#_Toc57559)

[**Generated Story** 42](#_Toc57560)

[**Installing Storyteller Master** 43](#_Toc57561)

[**Finalizing Attributes** 43](#_Toc57562)

[**Generated Story** 44](#_Toc57563)

[**Chapter 10; Evaluation** 45](#_Toc57564)

[**10.1 Measuring Quality** 45](#_Toc57565)

[**10.2 Further Planning** 45](#_Toc57566)

[**10.3 Story Telling Applications** 46](#_Toc57567)

[**10.3.1 Other Examples of Applications** 46](#_Toc57568)

[**Chapter 11; Discussions and Further Evaluation** 48](#_Toc57569)

[**11.1 Other Examples of Applications** 48](#_Toc57570)

[**11.2 Narratology Relevance** 49](#_Toc57571)

[**Chapter 12; Future Investigation** 50](#_Toc57572)

[**Chapter 13; Conclusion** 51](#_Toc57573)

[**Chapter 14; References** 52](#_Toc57574)

# Abstract

The goal of my dissertation is to create a Fictional Story Generator that will investigate the use of reader modeling to generate plots. There is a growing collection of literary and psychology research looking at how to interest readers in a story. A formal reader model can be used in the context of story production, although there are already systems in place that do so. These tools are mostly designed to produce interactive narratives, whereas the few that are meant to produce fixed narratives have mostly focused on the development of dialogue. Rather of plot creation, a predefined sequence of tale events is used. For this project, the goal will be to come up with and implement a new way for people to generate tales.

The project's goal is to create and execute this strategy. As a result, the tale generator will be constructed exclusively from user-generated material. The Engineering research concept was used to help the reader construct the Fictional Story Generator. Incorporating newer ideas from the field of reading psychology into everyday life should provide a fresh look at those theories themselves. Using the tool, you can also create a list of the animals' preferred diets and forms of exercise. As previously mentioned, a parser is utilized to represent all of the various knowledge bases and to integrate them into the user-driven narrative generator.

# Chapter 1; Introduction

The term Story Generator Algorithms (SGAs) refers to computational procedures resulting in an artifact that can be considered a story. Story generation with AI has been a topic for 50 years. [1, 2, 3, 4, 8].In an algorithm a group of instructions are applied to inputs, the it has an exact output. In this present situation, the expected output is a story. The idea of a tale is conceivable in SGA and excludes beautiful and ugly notations. [1, 4]. It is crucial because of the requirement for tales; in order to do this, they need be knowledgeable about surface realisation, and text appealing should not be a problem.

Making a movie or a video game would be impossible without a strong focus on story. Because they enhance the final product's charm. It is possible for youngsters to acquire vital insight into a topic when tales are employed as teaching tools. This means that in order to stay up with the present supply, new and improved materials must be regularly created [7]. Short tales from a wide range of writers are included in this anthology. Because of these new restrictions, small and mediumsized firms have asked the government for aid in complying.

One example of how computers may be used in the realm of information technology is the production of storytelling. There was a "happening," according to the accounts. "Story creation" or "story building" refers to the act of creating a tale from a collection of input qualities. With Over the last several decades, there has been a steady rise in the number of articles discussing the developments in artificial intelligence and processing power. Because of new generational algorithms being introduced to the industry, this was achievable. You may summarize a chunk of the current narrative by following this method.

As a further option, there are a variety of distinct generating algorithms.

1. The algorithm that is based on data.
2. The algorithm based on cases.
3. The algorithm based on roles.

## 1.1 System Aims

i. Generate Stories.

ii. Narrate Stories using Text to Speech.

iii. Download Stories as Text files.

iv. Get a globally sharable URL and QR code to share stories with anyone.

1. Get Suggestions for possible Grammatical/Spelling errors.

## 1.2 System Objectives

i. The investigation of existing algorithms for creating stores is now underway.

ii. The purpose is to execute an innovative technique for generating stories that are driven by the users themselves.

iii. Design and build a story generator built entirely out of user-generated content to maximize its effectiveness.

iv. It is important to satisfy the following criteria to evaluate the narrative generator based on feedback from users.

## 1.3 System Background

There is a gradual growth in story telling systems in recent years. Moreover, there is spike in developed systems in recent years. In the initial stage these systems are easy for humans rather than machines. To make the task easy for the machines humans developed AI.

There are some more other examples for this they are computer vision, speech processing and natural level language. From the above three examples the first two have achieved success and there is a spike in commercial applications. Natural level language and story generating systems are in still research stage. The problem is being assessed by the researchers.

## 1.4 Problem Statements

In story generation, generating a story is a difficult task and not well defined by the AI/computational Perspective. In any case, in order to develop an algorithm for a certain job, the inputs must be adequately specified in relation to the outputs.

In generating stories or fictional stories these are not defined properly. When humans construct tales, they invariably make the same mistake: "not providing adequate input.".

Furthermore, defining what constitutes a good tale is an unsettled subject.

## 1.5 System Scope

The story generating systems which are created previously are experimental in terms of more than simply the algorithm they employ, not just the qualities their intended output narratives should produce, but also the collection of inputs they start with. From all views and points of story generation or narratology, this has became an crucial one that different perspectives on these fundamental verdict goes to completely creation of new stories. The main benefits of this narratology is to study the concepts well.

## 1.6 Research Questions

1. How to be aware of the various story-generation algorithms available, as well as their respective limits, before writing?
2. Can system hold the user's attention while creating a new narrative?
3. How user-driven narrative creation method calls for the development and implementation of a story generator? iv. When using a user-driven story generating methodology, what is the primary purpose of the method in question?

v. Is testing the tale generator's output an option for ensuring its correctness?

# 1.7 System Functionality

i. Users can generate the story.

ii. User can access all past stories.

iii User can save the generated story in notepad.

iv. User can narrate the story in the speech.

## 1.8 Operating Environment

The product will be operating in any environment which has access to python IDEs. Major project development will be done on Python and will be done on different IDE so to operate this it just needs Python support. We can run the last fitted module on google collab also which is open source python IDE by Google.

# 1.9 Appropriated Technologies

i. Python

ii. Spyder

iii. Jupiter

# Chapter 2; Overall Description

All of the algorithms that may be used are listed here. Cons of each are that the story might become disjointed and that the narration has a propensity to repeat itself. The best way to create improvements is to become active in the community. It is the ultimate purpose of this study to aid in the creation of new technologies. Rather of relying on the user's input, they encourage them to take a more active part in the story-telling process [14].

By asking open-ended questions and paying great attention to the answers, this approach stimulates participation from the end user and fosters deeper levels of understanding. The storyline moves in the direction determined by the choices made by the participants [14, 15]. This approach has been fine-tuned over the course of several years. To the authors' credit, they based their conclusions on publicly accessible data, which they say may be obtained on the internet. Public information that has previously been made publicly the usage of bases in combination with other tools speeds up the product development process [15].

Everything boils down to this, as indicated in the study's conclusion. An example of the next generation's user-driven approach to storytelling might be shown by a short prototype application this user-driven narrative generator will be evaluated against other story generation tools already on the market through an online survey [15, 16]. This data was used to make observations and develop inferences. Users are in charge of running the tale generator, which shows how much more involved they are becoming in the project as time goes on. Tale interest is required for the user to write a story that is meaningful to them.

## 2.1 Existing Systems

In comparison to other story generators, this one is unique in that it exclusively uses fairy tales, allowing it to be more accurate. Existing story generators, especially those based on fairy tales will be examined in this research to see whether they can be utilized to produce new stories or if they can be improved upon. It will be much easier in the future to compare and evaluate the various narrative generators now that they are organized in this manner [16]. Two components that might be restricted to a certain level together with the domain and the background in order to accelerate growth [16, 17]. Despite its importance, the narrative generator's capacity to cover a wide variety of topics in this work is just a minor issue. When we finish this study, we want to know whether an innovative, user-driven method of building floor structures can be put to use in the real world.

The term story generating algorithms (SGAs) refers to computer techniques that result in a tale. Artificial intelligence (AI) researchers have been working on creating tales on their own for more than half a century. To put it simply, algorithms are the instructions that, when used in conjunction with one another, may generate a desired result.

A tale is what is needed in this situation. Functionally, "narrative" in SGAs does not entail any aesthetic idea. Setting the stage for evaluations of produced tales, which are not always predicated on a story's ability to be rendered as an understandable and visually beautiful text, is critical.

## 2.2 Other Systems

There are many different ways to convey a narrative out there. This review will focus on systems that produce typical sequential narrative. For various systems, examples of narrative output are provided for systems that have tiny enough meaningful pieces.

### 2.2.1 Novel Writer System

Sheldon Klein's Novel Writer technique is one of the first systems of storytelling [17, 18]. Novel Writer came up with murder mystery concepts for his books over a weekend party. According to others, this method can produce "2100-word murder mystery narratives with sophisticated structure" in only 19 seconds [18]. The author provided input on how the story should be depicted in addition to setting specifics like a character's personality attributes (which included emotional links between them and their predisposition towards violence or sex). The assassin and the victim were identified based on the information provided (with an additional random ingredient).

The story's events served as the catalysts for the protagonist's actions. A murderer's motives may only be categorized as greed, wrath, jealousy, or fear. Two algorithms were used to create the tale: one encoded future world changes using rules, while the other used scenes based on the kind of story being transmitted. As a result, both strategies were used concurrently. We have put in place some stringent guidelines to make sure that just one kind of story may be told. With this program, a story could be constructed in a variety of ways, but the only differences were in the techniques of murder and the individuals responsible for their discovery [19].

### 2.2.2 TALESPIN

TALESPIN (Meehan 1977) was a technique for generating tales about the lives of basic forest animals that was developed by Meehan. A character was assigned a goal, and then a strategy was devised to achieve the objective. TALESPIN used character ambitions as a means of eliciting action. It also opened the door to the story's problem-solving cast expanding too many members, each with their own set of objectives to work toward. Modeled were the complicated relationships between characters (competition, dominance, familiarity, affection, trust, deceit and indebtedness). A basic model of character motivation may be derived from these connections, which serve as prerequisites to certain behaviors and as outcomes of other activities. The characteristics of the characters were based on the degree of kindness, vanity, honesty, and intellect they had.

#### 2.2.2.1 TALESPIN Story Example

John Bear is a little bit hungry at the moment. John Bear is on a mission to get some berries for his breakfast. John Bear is eager to get close to the blueberries. From the cave to the bush, John Bear passes through a valley and into an open area of grassland on his way. The blueberries are John Bear's. John Bear devours the blueberries. The blueberries are all gone. John Bear is not hungry now.

### 2.2.3 Dehn’s AUTHOR

A software called AUTHOR (1981) by Dehn was designed to mimic the author's thinking while she creates a novel. Authors, according to Dehn, create fictional worlds, as post hoc justifications for actions they've previously determined would be included in the narrative. Even if an author does not have specific objectives in mind while writing a narrative, it is widely believed that the storytelling process is guided or constrained by a variety of Meta level objectives (Turner, 1994). It is important to make sure that the tale is logical, that the characters are realistic, that the reader's interest is held throughout the story, and so on (Michael, 1983). A sub goal of these may be the author's desire to place certain characters in certain circumstances, or to give them a certain role to perform in the plot (Michael, 1983). The "accomplishment of a complicated network of author aims" is how a tale is defined. These objectives aid in the story's organization and creation. There are no clear objectives in this story's conclusion [20].

### 2.2.4 Leobowitz’s UNIVERSE

Lebowitz’s UNIVERSE (1983) served as a blueprint for how to write a series of TV soap opera episodes in which a big ensemble of people performs several, concurrent, overlapping plots that never end [20]. It was in UNIVERSE that the concept of character creation was first introduced. A simple approach was devised to somewhat automatically fill up the data structures needed to represent characters. However, the majority of the characterization was left up to the user [20, 21].

#### 2.2.4.1 Dynamic Nature of UNIVERSE

Extending narrative creation, rather than a one-off tale, was the goal of the UNIVERSE project. In the beginning, it was meant to be a tool for writers, but subsequent plans called for it to take on the role of an independent storyteller. While making up a fictitious world, Universe addressed a matter of procedure: should the narrative drive the development of a world, or should the world be established first, with people, places and items being produced as necessary? Because of Lebowitz's preference for the first choice, UNIVERSE incorporated the ability to create characters without regard to the storyline. UNIVERSE's intriguing feature is that it alternated between producing a new episode to continue the plot and retelling the most recent episode it had previously made.

### 2.2.5 MINSTREL

MINT (Turner 1993) was an Arthurian-themed story-telling show that included tales of King Arthur and his Round Table companions. Each episode of the show was focused on a moral, such as "Deception is a weapon that is difficult to aim [21, 22]." MINSTREL's tales were between onehalf and one page long. To put it another way, the creator of MINSTREL claims that it can tell roughly 10 tales of this length, as well as a number of lesser ones.

MINSTREL assembled goals and plans into construction units in order to achieve them. They had two distinct purposes: authorial and characterial. Construction of tales in MINSTREL was a twopart process that included a stage of planning and a step of problem solving that made use of prior stories' expertise [22].

### 2.2.6 MEXICA

Computer model MEXICA (1999) was used to analyze the creative process. Short tales about the early people of Mexico were intended to be generated using this software application. A steady stream of fresh narrative material was created throughout the engagement period, with no restrictions in place. The created content was changed to meet general restrictions during the reflection phase.

#### 2.2.6.1 Story Example of MEXICA

Tenochtitlan's Great Tenochtitlan was home to the Jaguar Knights. Princess lived in Tenochtitlan, the largest city in the Aztec empire. When Ehecatl (the wind deity) blew, an ancient tree fell on Jaguar knight, seriously hurting him. After searching for medicinal herbs, Princess healed the Jaguar knight. Therefore, Jaguar knight was extremely thankful to Princess. The Jaguar knight gave Princess cacauatl (cacao beans) and quetzalli (quetzal) feathers as a token of gratitude for her bravery.

### 2.2.7 BRUTUS

The software BRUTUS (Bringsjord & Ferrucci 1999) wrote short tales on betrayal. BRUTUS was fascinating because it used a rational model of treachery to explain its tale. This model was able to provide a lot of interesting tales because of its complexity and the conclusions it can make. The system was also built to include a wide range of literary and grammatical skills.

BRUTUS wrote one of the most stunning stories I have ever read. It included a wide range of literary elements, including literary tropes, dialogue, character identification, and more. Though it may seem to be an original piece of work by the writers, BRUTUS is really the outcome of an experiment to test if it can generate a tale from scratch.

### 2.2.8 FABULIST

Automated narrative production and presentation was made easy using FABULIST (Riedl & Young, 2010). There were three stages of fabula creation, discourse formation, and media representation in Fabulist architectural design. In the fabula creation process, a planned approach to story generation was applied. Given a description of the current state of the world, and a specified objective, AI planners may discover the most efficient path to achieve that goal. Preconditions and post conditions of every activity are described in great depth. Narrative creation is planned around the idea that a series of events moving from a starting point to a destination is a reasonable approximation of a tale. Inputs for FABULIST comprised a model of the tale world's beginning state, actions that characters may take and a result.

## 2.3 Explications

When it comes to computer systems that can tell stories, there is a lot of interest in the field. The number of systems that have been created in the last few years has expanded dramatically [23] Because of the broader tendency in AI to develop, computational solutions that could do activities that are simple for people but difficult for computers, these systems originally emerged [24]. Computer vision, voice processing, and natural language comprehension are some instances of this development. There have been commercial uses for just the first two instances. Natural language processing and narrative production are yet in the exploratory stages of study [25].

There is a lot of room for improvement when it comes to narrative development since it is not clearly defined from an AI or computational standpoint. The inputs and output properties of an algorithm should be crystal apparent if it is to be developed for a specific goal. There is no precise definition of any of these in the creation of tales [25, 26]. Often, when people are involved in the creation of tales, it is difficult to tell exactly what they are contributing to the final product [25]. Furthermore, the definition of what constitutes a good tale is still up for discussion. This has resulted in a wide range of current narrative generating systems that are exploratory in terms of both their algorithms and the set of inputs they begin with, and the traits they hope to create. There are varieties of approaches to narratology that may be derived from various perspectives on these essential choices [27]. Because of SGA study, narratological notions are better understood.

# Chapter 3; Literature review

There are multiple tale generators now available, and this section provides a brief overview of each one and an explanation of the techniques they utilize [1]. Algorithmic narrative generators will be compared and contrasted. In order to examine the relative merits and drawbacks of each generator [2]. An important component of this project is the creation of a public ontology that will serve as the basis of a knowledge base for the tale generator. Many businesses, including game development, education, and the entertainment industry, have the potential to gain from storytelling [3]. There are several tale generators on the market today, and this section will detail the different ways each use to produce their stories.

Among the great majority of storytellers, the databases and ontologies they depend on serve as their “basic knowledge” [3, 4]. By just browsing through this database, you may create new sentences from a large number of the terms. To provide one example of how ontologies may be used, they are already being utilized in a variety of tale generators [4]. By using ontology, connections between things may be represented in a more realistic way. The philosophical word "ontology" has its origins in metaphysics, although it has subsequently gained acceptance in a broad range of philosophical traditions [5]. In ontology, the underlying links between concepts like "being," "existence," and "reality" are investigated in more depth.

By definition, ontology refers to the study of things at their most fundamental level. Definition: As described in "Tom Gruber ontology," an explicit and formalized statement of notion, it is defined as such in Tom Gruber's book [5, 6]. Ontologies, a sort of database, are used to record information on the connections between different structures and pieces of information. In addition, they may be used to keep track of things like entities, activities, and other relevant information (Obitko "Ontologies and Semantic Web") [6]. In the field of computer science, an ontology is an organized collection of information on a certain topic that has been categorized using classification. It is possible to categorize things, ideas, and interactions between them in our environment using a wide range of diverse categories [7]. All of these are useful for a number of reasons. This information may be found in the ontology, which is expressly indicated in the text where it was found [8]. An ontology enables a domain to be more precisely specified and characterized, and therefore, it may be more precisely defined.

## 3.1 Literature Background

### 3.1.1 GPT 3

GPT-3 made the mainstream media headlines this year, generating far more interest than we would normally expect of a technical advance in NLP. People are fascinated by its ability to produce apparently novel text that reads as if a human wrote it [14]. The mid-year release of Open AI’s GPT-3 language model, with its ability to generate natural language texts that can be remarkably hard to distinguish from human-authored content, was this year’s big AI news item [5].

### 3.1.2 Long Text Progressive Generation

Large-scale language models (LMs) retrained on massive corpora of text, such as GPT-2, GPT-3, are powerful open-domain text generators. However, as our systematic examination reveals, it is still challenging such models to generate coherent long passages of text (e.g., 1000 tokens), especially when the models are fine-tuned to the target domain on a small corpus [13]. Previous planning-then-generation methods also fall short of producing such long text in various domains. To overcome the limitations, we propose a simple but effective method of generating text in a progressive manner, inspired by generating images from low to high resolution [3].

### 3.1.3 Controllable Text Generation

Controllable Text Generation (CTG) is emerging area in the field of natural language generation (NLG). It is regarded as crucial for the development of advanced text generation technologies that are more natural and better meet the specific constraints in practical applications [12]. In recent years, methods using large-scale pre-trained language models (PLMs), in particular the widely used transformer-based PLMs have become a new paradigm of NLG, allowing generation of more diverse and fluent text. However, due to the lower level of interpretability of deep neural networks, the controllability of these methods need to be guaranteed [3, 12]. To this end, controllable text generation using transformer-based PLMs has become a rapidly growing yet challenging new research hotspot.

### 3.1.4 Plug and Play Language Model

Large transformer-based language models (LMs) trained on huge text corpora have shown unparalleled generation capabilities [11]. However, controlling attributes of the generated language (e.g. switching topic or sentiment) is difficult without modifying the model architecture or fine-tuning on attribute-specific data and entailing the significant cost of retraining.

# Chapter 4; Software Requirements Analysis

## 4.1 Performance Requirements

Some performance requirements are developed during the development of the system for the effective performance of the system as well for the better results in efficiency.

i. The system shall accommodate different users without any failure and Faults.

ii. Response to user requests or information gain shall take no longer than 5 seconds to appear on the screen.

iii. The system should be available to the user.

iv. The system shall be accessed using any python open source.

v. If required we can link this to Open web pages later on.

## 4.2 Safety and Security Requirements

i. The system will be protected end-to-end.

ii. The system will use a secured database.

iii. The system must be private as we are aiming at a private story generator.

iv. The running shall not be connected to any other environment.

v. The system use shall not harm the user in any mental or health way.

## 4.3 Software Quality Attribute

### 4.3.1 Extensibility

The system can be extended later with other required functionalities according to user.

### 4.3.2 Portability

The system can be extended later with the web application is based on different languages (HTML, scripting languages).

### 4.3.3 Maintainability

The system uses secure database for maintaining the data. In case of any failure, the program will be re-initialized instantly. The architecture of software design is designed in modules keeping in mind that maintainability of the system should be carried out efficiently.

### 4.3.4 Security

The system uses SSL (secured socket layer) Certificates. In addition, all confidential information is protected end-to-end. The system can be designed on sessions, as a user will be logged out automatically after several time of inactivity (Meehan, 1981).

### 4.3.5 Usability

The system shall have friendly command line user interface and user shall interact easily. The system shall not be designed complicated on user’s end.

### 4.3.6 Reliability

The system shall be designed to extent of failure-free system. In addition, if in case of failure it shall recover in no time maintaining the previous log.

## 4.4 Quality Assurance

Quality assurance (QA) is a way of preventing mistakes and defects in manufactured products and avoiding problems when delivering products or services to customers; which ISO 9000 defines as "part of quality management focused on providing confidence that quality requirements will be fulfilled.

### 4.4.1 Introduction

The main aim of testing is quality assurance. Whatever technique is being used, analyze the performance and to evaluate the errors that occur when the program is executed with different input sources and running in different operating environments (James, 2002). The main aim of testing in this project is finding defects, prevent them, and gain confidence of the stakeholders by providing them a quality product (Selmer, 1999).

# Chapter 5; Methodology

This inquiry will use the Engineering Research Methodology to obtain data for the objective of this investigation and system. These actions need to take:

1. Consider some of the concepts that are being explored for execution in this research.
2. Many story generator systems are available today, each with its own set of benefits and drawbacks to weigh.
3. Collect and analyse the data of GPT.
4. Can this model handle usual story generators?

A technique provides guidance, saves a lot of time, and helps to concentrate on the most relevant components of a study. There are several approaches to choose from, and doing so wisely is critical to the success of this investigation. The following sections provide an explanation of the research's selected technique of analysis.

## 5.1 Engineering Research Methodology

This inquiry will use the engineering research approach to obtain data for the objective of this investigation and system. These are the actions that you will need to take: Consider some of the concepts that are being explored for execution in this research. Many story generator systems are available today, each with its own set of benefits and drawbacks to weigh.

## 5.2 Model View Controller Architecture

Our system is designed according to standard code design principles. When it comes to its capacity to accomplish a broad range of duties. It will be employed in a wide range of applications and scenarios. Because of the system’s efficiently designed principles and structure.

## 5.3 Methodology Investigation

Reviewing relevant literature was part of this inquiry, which aimed to better comprehend the generators under discussion. In order to evaluate and contrast the many tale generators that are currently available in the previous phases. A wide range of criteria will be used, and the findings will be presented in the sections that follow. When comparing the different criteria, take into account how much time will be spent speaking with people, what sort of knowledge base tools were utilized, how effectively the language is organized, and whether or not the system can go up the ladder from one level to another.

Because of the variables listed above, it is feasible to analyze, evaluate, and compare the tales created by various generators on a wide range of factors. Including, but not limited to: a wide range of aspects an additional goal for this project is to produce a tale generator that is more user-friendly in the natural setting, which is one of its other aims in addition to this.

## 5.5 Research Methodology Plan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | DURATION 1 DIV = 1 WEEK | | | | | | | | |  |  |  |  |  |
| ACTIVITY | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 |
| Identification of supervisor and project proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Collecting the data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Literature survey |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Study the Expansion Joint from all aspect |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Preparation of Report  proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Final submission |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Chapter 6; Approach

**NLG**

NLG and NLP are the two terms that are used frequently when we talk about considering a text in any of the Machine Learning or deep learning problems. Most machine learnings are designed to take mathematical data as input for the model. The name itself stands for Natural Language Generation which says it is the process of generating natural language.

Natural language generation is one of the best applications of artificial intelligence developed to date, it helps with a lot of human problems. Everyone must have interacted with some or the other online or offline available chatbots which is one of the biggest things using natural language generation in the backend.

NLG is the method involved with changing information into regular language utilizing man-made consciousness.

NLG programming does this by utilizing man-made consciousness models fuelled by AI and profound figuring out how to transform numbers into regular language text or discourse that people can comprehend.

Chatbots, voice colleagues, and man-made intelligence blog essayists (to give some examples) all utilize normal language age. NLG frameworks can transform numbers into accounts in light of pre-set formats. They can foresee which words should be created straightaway (in, say, an email you're effectively composing). Or on the other hand, the most modern frameworks can plan whole synopses, articles, or reactions.

or the most part, correspondence is one of the significant things that people required. Much data can be separated through a site or web with for the most part in regular language. The most issues while getting that data are equivocal, muddled words, or casual sentence structure.

How Natural Language Generation works, basically it is not a single-step process. It involves AI so it is a lengthy and very complicated process. Which involves steps Like Analysis of the Content, Base Understanding of the Content given, Main document structuring, Aggregating of the sentence, structuring grammar, and presentation of language. So basically, these are six important stages of natural language generation. Rest the complexity of the complete process depends on the application we have and the variety of data we have in our dataset.

In today’s world where everything is going digital day by day, the use of AI in text generation is increasing day by day. Some of the real-time applications of Natural Language Generation, Chatbots where you are talking to a computer and it generates a response to everything you ask, one of the other biggest applications of this is automating emails, messaging, or chat responses. Many of the biggest use automatic responses and automatic response generation these days which in the end helps them to reduce the manual tasks they have, when a response is generated by a computer or ai it saves human efforts which ate the end helps any organization. Apart from these many organizations use personalized messaging and scribing bots for generating mass emails and messages.

Natural Language Generation models and methodologies, there are a few major techniques to do this four are the most important and common methods which are being used for this, Markov Chain is a category of the mathematical model used in statistical applications to understand and analyse different things. RNN Recurrent Neural Networks are also widely used for natural language generation, majorly these are used where we have a lot of sequential data because neural networks are designed in a way to handle such data properly. LSTMs and transformers are also used in NLG where LSTM is used where the model needs to learn from the experience model gains and transformers are for long range and more dependencies.

**NLP**

Natural Language Handling (NLP) is one computer-based intelligence that spotlights the human normal dialects to peruse, comprehend, or separate their importance of it.

Normal language is a language that for the most part utilized by people for correspondence. Contrast with the PC that needs to handle before grasping it.

By definition, normal language is one structured message that permits communicators (source) to impart to communicants (beneficiaries) through a medium (like sound or message). NLP can be deciphered as a parser sentence that peruses the sentence, word by word, and decided the following kind of word.

**Area of Interest**

There are a few areas of interest in creating,

**Question Answering Systems (QAS) -** QAS provides the capacity to address questions that the client gave. Rather than looking through the web search tool by watchword, the client can straightforwardly ask in regular dialects.

**Summarization**- As the named, outlining a bunch of content. The application can assist clients with changing over huge text archives into more modest structures without eliminating significant data.

**Machine Interpretation** - As far as we might be concerned can cause the machine to comprehend people's language and then make an interpretation of it into another dialect. For instance, Google Decipher.

**Discourse Acknowledgment** - This field is an unobtrusively troublesome part of NLP. All things considered, a few models for phones or PC utilizes currently a great deal.

**Record Characterization** - This is for the most part utilized these days. The application can decide the inputted reports in the framework to where should the archive be put. We can meet it in spam sifting, news story characterization, or even book surveys.

Natural Language Handling is a piece of man-made consciousness that intends to show the human language with every one of its intricacies to PCs. This is so that machines can comprehend and decipher human language to comprehend human correspondence in a superior manner ultimately. Normal Language Handling is a cross among various fields like man-made reasoning, computational etymology, human-PC collaboration, and so forth. There is a wide range of strategies in NLP to comprehend human language which incorporates measurable and AI techniques. These include separating human language into its most essential pieces and afterward comprehending how these pieces connect and cooperate to make implications in sentences.

Also, for what reason is Natural Language Handling significant, you wonder? Indeed, it permits PCs to comprehend human language and afterward break down immense measures of language-based information in an unprejudiced manner. This is extremely challenging for people to achieve. Also, there is a large number of human dialects in many vernaculars that are spoken in various ways by various ways. NLP helps settle the ambiguities in language and makes organized information from an extremely complicated, jumbled, and unstructured source.

This is the explanation that Natural Language Handling has numerous different applications nowadays in fields going from IT to broadcast communications to scholastics. In this way, how about we see these applications now?

Utilizations of Natural Language Handling

1. Chatbots

Chatbots are a type of man-made reasoning that is customized to communicate with people so that they sound like people themselves. Contingent upon the intricacy of the chatbots, they can either answer explicit watchwords or they might hold full discussions that make it intense to recognize them from people. Chatbots are made utilizing Regular Language Handling and AI, and that implies that they comprehend the intricacies of the English language and find the genuine importance of the sentence they likewise gain from their discussions with people and become better with time. Chatbots work in two basic advances. In the first place, they recognize the significance of the inquiry posed and gather every one of the information from the client that might be expected to address the inquiry. Then, at that point, they answer the inquiry suitably.

2. Autocomplete in Web search tools

Have you seen that web indexes will more often than not think about the thing you are composing and naturally complete your sentences? For instance, on composing a "game" in Google, you might get further ideas for "round of privileged positions", "round of life" or on the other hand if you are keen on maths, "game hypothesis". This multitude of ideas is given utilizing autocomplete that utilizes Regular Language Handling to think about what you need to inquire. Web search tools utilize their huge informational collections to dissect what their clients are likely composing when they enter specific words and recommend the most well-known potential outcomes. They utilize Natural Language Handling to figure out these words and how they are interconnected to frame various sentences.

3. Voice Partners

Nowadays voice partners are the fury! Whether it's Siri, Alexa, or Google Right hand, nearly everybody utilizes one of these to settle on decisions, place updates, plan gatherings, set alerts, surf the web, and so forth. These voice collaborators have made life a lot simpler. In any case, how would they work? They utilize a mind-boggling mix of discourse acknowledgment, normal language understanding, and regular language handling to comprehend what people are talking about and afterward follow up on it. The drawn-out objective of voice collaborators is to turn into an extension among people and the web and give every kind of administration because of simple voice communication. Nonetheless, they are still somewhat distant from that objective seeing as Siri actually can't comprehend what you are talking about once in a while!

4. Language Interpreter

Language interpreters are mostly used to convert or translate one language into the other. These days google provides you with most of the services for free, google also has a lot of language interpreters most of them are available for free to everyone, so whenever your phone or Internet translates something google uses a language interpreter in the backend. Google Decipher and other interpretation devices as well as use Succession to a group demonstrating that is a method in Normal Language Handling. It permits the calculation to change over a grouping of words starting with one language and then onto the next which is interpretation. Prior, language interpreters utilized Factual machine interpretation (SMT) which implied they examined a great many records that were at that point interpreted starting with one language and then onto the next (English to Hindi for this situation), and afterward searched for the normal examples and fundamental jargon of the language. Nonetheless, this technique was not that exact when contrasted with Arrangement to grouping displaying.

5. Opinion Examination

Practically all the world is via web-based entertainment nowadays! Furthermore, organizations can utilize opinion investigation to comprehend how a specific kind of client feels about a specific point, item, and so on. They can utilize normal language handling, computational semantics, text examination, and so on to comprehend the overall opinion of the clients for their items and administrations and see whether the feeling is great, awful, or impartial. Organizations can involve feeling examination in a ton of ways, for example, to figure out the feelings of their main interest group, to comprehend item surveys, to measure their image opinion, and so on. What's more, not simply privately owned businesses, even states use feeling examination to find prevalent sentiments and get out any dangers to the security of the country.

6. Language structure Checkers

Language structure and spelling is a vital variable while composing proficient reports for your bosses and even tasks for your teachers. All things considered, having significant mistakes might get you terminated or fizzled! That is the reason sentence structure and spell checkers are vital devices for any expert essayist. They might not just right syntax and actually, look at spellings at any point yet additionally recommend better equivalents and work on the general clarity of your substance. Furthermore, think about what, they use regular language handling to give the most ideal piece of composing! The NLP calculation is prepared on a huge number of sentences to grasp the right organization. To that end, it can propose the right action word tense, a superior equivalent, or a clearer sentence structure than what you have composed. Probably the most famous punctuation checkers that utilize NLP incorporate Grammarly, WhiteSmoke, ProWritingAid, and so on.

7. Email Arrangement and Sifting

Messages are as yet the main strategy for proficient correspondence. Notwithstanding, we all receive a large number of limited-time Messages that we would rather not read. Fortunately, our messages are naturally partitioned into 3 segments to be specific, Essential, Social, and Advancements which implies we never need to open the Limited time segment! Yet, how does this work? Email administrations utilize regular language handling to recognize the items in each Email with text characterization so it tends to be placed in the right segment. This strategy is noticeably flawed since there are still a few Special pamphlets in Essential, however, it is not great, but not terrible either than nothing. In further developed cases, a few organizations likewise use specialty hostile to infection programming with regular language handling to check the Messages and check whether there are any examples and expressions that might show a phishing endeavour on the representatives.

**How NLP Natural Language Generation Works?**

NLP Natural Language Generation helps most of the digital world users on daily basis. Most of the Natural Language Processing model relays on Deep Learning and its algorithms. Most of the model use past human data to understand the patterns in one way or the other and then use those patterns or understanding as the base for future work or predictions. You must have seen devices like Hey Google, Alexa, and Siri all of these NLP to understand a process.

The complete process of Natural language Processing includes many steps, it is not a one-step process, as we cannot process text data into deep learning models so the complete process converts the data into equivalent mathematical data. So, the process is divided into the following steps.

A. Basic Data Filtering - This step includes all the basic data filtering processes like removal of duplicate data, removal of Null data, etc.

B. Tokenization - These steps include dividing the complete sentence into the bag of words by considering the main words, this step removes all special symbols we have in our sentences.

C. Stop Word Removal - This step is all about taking those words out which has zero polarity Like How, Where, too, etc. These words have zero polarity. These words are highly frequent, words that have less weight in any term are known as stop words. So, in this step, we remove all these and convert the bad words into another bag of words that has all meaningful words.

D. Stemming/ lemmatization - These steps are used to convert the words to their root form because Root words and the other higher-form words have the same context and we are only considering the context of the word. So, converting the higher form of words into the root forms helps us to remove all separate words with one single word, which in the end reduces the unique number of words we have, which makes learning and finding patterns easy.

E. Vectorization - As the machine and deep learning models need mathematical data input. So, this converts the final bag of words into mathematical matrices which can be considered good inputs for models. One of the most commonly used vectorizers is the Count Vectorizer, which converts the text data into vectors based on the frequency of words in the entire text.

**Story Generation**

We could recognize the mechanized story age and robotized plot age. As of late, some have begun distinctive story age from plot age as whether the result of the framework peruses as a blueprint of headliners as opposed to having normal language that portrays parts of the story that are not rigorously occasions, similar to depictions, discourse, and different elaborations. The differentiation is by all accounts: does it read like an undeniable level layout of occasions, or does it seem to be something somebody could track down in a book? I'm unsure I would make this qualification, but rather I see the interest for those that are keener on surface structure versus structure (two similarly significant and legitimate points of view).

The made-up rules separate mechanized story age from other narrating advancements, for example, news composing, where the occasions are those that truly occurred in reality. That is, news depends on this present reality as the "generator" and afterward makes a characteristic language writing. News age is a significant issue, yet one that, as I would see it, ought to be isolated from robotized story age.

Negligible information is a rule that I add to mechanized story age to recognize from story retelling. Story retelling is an issue where most of the story/plot is all given and the computerized framework is delivering a result that tracks the information intently. For instance, one could give a story-retelling framework a hint of realities about a story in some contracted or organized structure and the framework could produce normal language composition that conveys those realities altogether. For this situation, the "story" was at that point known however the surface type of the telling is variable. Concerning what an "insignificant" set of info is liable to discuss. Would it be a good idea for it to be a solitary provoke for the beginning of the story? Would it be a good idea for it to be the beginning and an objective? Would it be a good idea for it to be a few plots focuses that get filled ready? And given space information? For AI frameworks, would it be a good idea for us to consider the corpus that is prepared as info? This presumably needs more work however I'm disinclined to give a definition that is excessively restricted.

**GPT**

GPT stands for Generative Pre-Trained Transformer. GPT is a pre-trained deep learning model which is trained on large human text data to produce human-like text, the way humans speak words. This model is trained on a very large dataset which helps this to generate nearly the same way how human acts in some situations. The main design of this model is based on the standard design of transform or transformers networks. The complete data used for this model has nearly 175 billion parameters used. The model is used for regressive used to like it is used to predict the next move, example if you are writing a sentence and you are in between writing that then this model can help you to predict what was you writing.

The quality of the text generated by this model is so good that you can differentiate between the text generated by this model and the text's usual human writing anywhere on a certain topic or with some context.

Open AI, is the term that helps a lot of artificial enthusiasts while building any application. Open ais are the models or algorithms which are available on open platforms and can be used by anyone according to whatever need they have. building a complete text generator is very difficult but just because of Open AI many of the models are openly availed on google completely free of cost to use, you just need the data set and the code to use it, and both of them are visible on the google.

The latest model of GPT has data which has almost sixty percent of tokens from Common Web Crawling data the count is nearly 410 billion, apart from this one of the major token bases are collected from WebText2, nearly twenty-two percent which is almost 19 billion tokens are taken from this source of data. Rest all the tokens are collected from books data and Wikipedia.

When GPT was developed almost 32 Engineers including a large team of researchers worked to develop this. Also, when the team designed its third generation which is also known as GPT-3, they increased the magnitude and capacity by two orders, which makes the latest GPT almost 300 percent stronger than the last one, but again great accuracy comes with a large dataset. As we all know that the amount of data we have for any machine or deep learning model is directly related to the accuracy we will going to have, if the dataset has the same principal concept, so to the point we have to take care of the aim we have the amount of data we have as input is directly proportional to the accuracy we will going to have. Especially in terms of text generation, the more amount of data we have to consider for the model gives the more strength. So as the generation of GPT is moving ahead the model is trained on more and more data, so by the time the data is increasing, the model gets good and becomes more similar to human behaviour.

The main application of the GPT model is generating text, code, and stories, as the model has different datasets for different applications, like if you want to generate horror stories, so for that you need GPT, but in this, you do not need the complete dataset you just the segment of data which is useful for you.

At Narrative, content robotization is a principal move toward accomplishing total computerized change. As a trailblazer in creating news through computerized reasoning, Narrative has acquired a presence in the media. In any case, this isn't the main area that can profit from this sort of innovation. The recent update of GPT which is GPT-3 was introduced in May 2020, which is still new because it is still under progress it is not fully availed.

**Parts of Code**

**1 > Basic GPT Run**

So here I ran a simple GPT model to check the basics of how all of this works.

**2 > Greedy Search**

When the process of story generation is taken into consideration, we mainly consider three different ways of decoding. Greedy Decoding, Random, and Beam Decoding, we also call them searching algorithms. So here we will be going to use two of them one is the greedy search which is the simplest decoding technique the other is the beam search. We always think that most of the words any model pick is because of the probability or the frequency of that particular token but it is not true, many models work not on a single token but on different probabilities of different words and then pick according to the context we have.

A greedy algorithm chooses the best solution that is currently available to solve a problem. It is unconcerned with whether the current best result will provide the final, ideal result. Even when a judgment is made incorrectly, the algorithm never goes back and changes it. It functions from the top down. For all the problems, this algorithm might not yield the optimal outcome. It does this because it consistently chooses the option that will result in the best global outcome.

The greedy algorithm is the simplest way of decisions we have, this method usually helps us to search the word according to the probability we have, in the set of probabilities we have given inside this according to the Argmax. But one of the biggest issues of greedy search is that it fails on long text. Also, when we use this to get a long text it starts repeating the same value an, so this failed when we need long text to be generated, as this is not preferred when you want more than 4 or 5 words sentences.

Also, there we go: it is that simple to create text. Our outcomes are not perfect - as may be obvious, our model beginnings rehashing the same thing rather rapidly. The main pressing concern with Covetous Hunt is that words with high probabilities can be veiled by words before them with low probabilities, so the model can't investigate more different mixes of words. We can forestall this by executing Beam Search.

**3 > Beam Search**

A heuristic procedure is a bunch of measures for figuring out which of the different choices will be the best for accomplishing a specific objective. This procedure builds the productivity of an inquiry interaction by giving up cases of efficiency and culmination of the best.

We can expect to accomplish a decent answer for troublesome issues (like the mobile sales rep issue) in under example time on the off chance that we utilize fitting heuristics.

A heuristic pursuit calculation that looks at a diagram by broadening the most encouraging hub in a restricted set is known as shaft search.

Shaft search is a heuristic hunt procedure that generally grows the W number of the best hubs at each level. It advances level by level and moves downwards just from the best W hubs at each level. Shaft Search utilizes expansiveness first pursuit to construct its inquiry tree. Pillar Search builds its inquiry tree utilizing expansiveness first pursuit. It creates every one of the replacements of the ongoing level's state at each level of the tree. Be that as it may, at each level, it just assesses a W number of states. Different hubs are not considered.

The heuristic expense related to the hub is utilized to pick the best hubs. The width of the pillar search is meant by W. If B is the spreading factor, at each profundity, there will continuously be W × B hubs viable, yet just W will be picked. More states are managed when the pillar width is diminished.

At the point when W = 1, the pursuit turns into a slope-climbing search in which the best hub is constantly browsed the replacement hubs. No states are pruned on the off chance that the shaft width is limitless, and the pillar search is distinguished as an expansiveness first pursuit.

The beamwidth limits how much memory is expected to finish the inquiry, yet it comes at the expense of fulfilment and optimality (conceivably that it won't track down the best arrangement). The justification for this risk is that the ideal state might have been pruned.

Now that is vastly improved! The 5 different pillar speculations are no different either way, however, if we increased num\_beams, we would see some more variety in the different bars. Pillar Search is somewhat flawed by the same token. It functions admirably when the length of the produced text is pretty much steady, similar to issues in interpretation or rundown, however not such a huge amount for unassuming issues like an exchange or story age (since it is a lot harder to track down a harmony among num\_beams and no\_repeat\_ngram\_size)

Besides, research shows that human dialects don't follow this 'high likelihood word next' dissemination. This seems OK - on the off chance that my words were the precisely exact thing you anticipated that they should be, I would be a seriously exhausting individual and a great many people would rather not be exhausted!

**4 > K/P Sampling**

Causal language models like GPT-2 are prepared to anticipate the likelihood of the following word given some specific situation. For instance, given "I ate a heavenly hot \_\_\_", the model might anticipate "canine" with 80% likelihood, "flapjack" with 5% likelihood, and so on. The cool thing about this construction is it can be utilized to create arrangements of erratic lengths. I can give the model "I ate," test a token from the subsequent dispersion to get "I ate a", then, at that point, put that through the model again to get another conveyance and coming about the token. Rehash as long as we like. It just so happens, this age frequently either stalls out in dull circles or fails to remember the subject and goes off point. For what reason is this occurrence, and how should we better example to create more human-like text?

This post is an outline and investigation of The Inquisitive Instance of Brain Text Degeneration by Holtzman et al 2019. I found it perhaps the most exhaustive and comprehensible paper I've perused in ongoing memory, so kindly look at it if this post arouses your curiosity!

Assuming we generally test the most probable word, the standard language model preparation objective makes us stall out in circles as "I don't have any idea. I don't have any idea. I don't have any idea." This is unnatural, however, the majority of the model's consideration in current language models is just on the latest few tokens. All things considered, famous testing techniques for age depend on examining the appropriation. Yet, examining likewise runs into an issue: assuming we have 50K potential options, regardless of whether the base 25K tokens are each very far-fetched, in total they could have for instance 30% of the likelihood mass. This implies with each example; we have a 1 out of 3 opportunities to crash our "line of reasoning." As a result of the short setting referenced before, this will cause an unrecoverable blunder overflow as each next word relies vigorously upon this new off-base word. To battle testing from the tail, the most famous strategies are temperature and top k examining. Temperature inspecting is enlivened by factual thermodynamics, where high temperature implies low energy states are almost certainly experienced. In likelihood models, logits assume part of the energy and we can execute temperature examination by isolating logits by the temperature before taking care of them into SoftMax and acquiring our testing probabilities

In Top-K examining, the top k no doubt next words are chosen and the whole likelihood mass is moved to these k words. So rather than expanding the possibilities of high-likelihood words occurring and diminishing the possibilities of low-probability words, we simply eliminate low-likelihood words generally together

Top-P testing (otherwise called core examining) is like Top-K, yet rather than picking the top k undoubtedly words we pick the littlest arrangement of words whose all-out likelihood is bigger than p, and afterward, the whole likelihood mass is moved to the words here

The fundamental contrast here is that with Top-K examining, the size of the arrangement of words is static (clearly) though, in Top-P testing, the size of the set can change. To utilize this examining strategy, we just set top\_k = 0 and pick a worth top\_p

As you might have most likely speculated, we can utilize both Top-K and Top-P inspecting here. This diminishes the possibility of us getting bizarre words (low likelihood words) while considering a unique determination size. We want just top an incentive for both top\_k and top\_p. We might incorporate the initial temperature boundary assuming we need to, how about we presently perceive how our model performs now after adding everything together? We will take a look at the best 5 re-visitation of perceiving how different our responses are:

**We checked the following conditions**

**1 > Can help to do student homework?**

So here we tried the different models to try to generate different stories based on their dataset, this is just to check if the model is capable to handle such situations or not.

**2 > Can generate fake news?**

In this, I passed one sentence which sets the context for fake news and the model responds well to this as well, so the model is capable to generate fake news as well.

**3 > Can generate horror stories?**

In this, I passed one sentence which sets the context for horror stories and the model responds well to this as well, so the model is capable to make horror generations as well we can also state that this model is good at building every category of story.

## 6.6 Development Plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Approaches** | **10 Days** | **10-20** | **20-30** | **30-40** | **40-50** | **50-60** | **60-70** | **70-80** |
| **Requirements**  **Gathering** |  |  |  |  |  |  |  |  |
| **Selection of**  **Approach and Design** |  |  |  |  |  |  |  |  |
| **Coding** |  |  |  |  |  |  |  |  |
| **Testing /**  **Execution** |  |  |  |  |  |  |  |  |
| **Final**  **Upgradation** |  |  |  |  |  |  |  |  |

# Chapter 7; Gantt chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity  Number | Task Name | Duration | Start  2022 | End  2022 | Month  1 | Month  2 | Month  3 | Month  4 | Month  5 | Month  6 |
| 1 | Project  Management | 4 Months | 01  April | 31 Aug |  |  |  |  |  |  |
| 2 | Project  Design | 2 Months | 01  July | 31 Aug |  |  |  |  |  |  |
| 3 | Define  Features | 10 Days | 10 May | 20 May |  |  |  |  |  |  |
| 4 | Code  Implement | 4 Month | 01 May | 25 Aug |  |  |  |  |  |  |
| 5 | Setup  Environment | 10 Days | 05  July | 15  July |  |  |  |  |  |  |
| 7 | Code / Setup | 2 Week | 21  June | 05  July |  |  |  |  |  |  |
| 8 | Testing | 2 Weeks | 20 May | 2 Aug |  |  |  |  |  |  |
| 9 | Fixing Bugs | 2 Weeks | 21  July | 05 Aug |  |  |  |  |  |  |
| 10 | Closure | 2 Days | 31 Aug | 01  Sep |  |  |  |  |  |  |

# Chapter 8; Analysis

Several different kinds of algorithms are used to create the tales produced by the systems examined above. It is not uncommon for a single algorithm to be used in many systems. There are several sorts of algorithms, and each is explained in detail in this portion of the article.

**We checked the following conditions**

**1 > Can help to do student homework?**

So here we tried the different models to try to generate different stories based on their dataset, this is just to check if the model is capable to handle such situations or not.

**2 > Can generate fake news?**

In this, I passed one sentence which sets the context for fake news and the model responds well to this as well, so the model is capable to generate fake news as well.

**3 > Can generate horror stories?**

In this, I passed one sentence which sets the context for horror stories and the model responds well to this as well, so the model is capable to make horror generations as well we can also state that this model is good at building every category of story.

# Chapter 10; Evaluation

Experimentation should show that my reader model has a significant impact, and should be able to identify which limitations play a role. Short produced tales (such as "was character A furious") may be used to test different parts of the reader model by asking people to score certain traits (e.g. a portion of the reader model that attempts to mimic inferences about emotion). These features are not impossible to uncover. However, they are more difficult to quantify in terms of narrative coherence or a central theme.

1. A reader model's value may also be assessed by applying the model to previously published tales.
2. The reader model will be asked to point out errors in both unmodified and amended versions of the tales that I transform into the logical representation of the system.
3. It is desirable that the reader model's predictions match with a human examination of a fixed tale. iv. Provide reasonable ideas for changing a story where a crucial event has been omitted deliberately.

v. This project will act as a mental health check and guide future system development.

## 10.1 Measuring Quality

The overall quality of system output will be assessed via the use of a survey technique that measures narrative engagement. I will be comparing generated tales to human-authored stories based on produced plots and human-authored stories. Also the output time of the system over the defined range of words. Achieving parity with human-produced narratives is unlikely. However, I believe it is critical to provide a realistic baseline for the generated narratives.

## 10.2 Further Planning

Qualitative assessments, in which I ask participants to provide more specific comments, are also in the works. When creating the system, we will utilize this qualitative input mostly for finding problems with the created tales. I intend to defend the reader-model-based approach not just by proving particular benefits but also by demonstrating the overall quality of the system's output by releasing this qualitative input.

## 10.3 Story Telling Applications

Besides traditional storytellers, there has been a massive increase in interactive storytelling apps. As the name suggests, these are interactive computer programs that let the user to control the actions of an individual character in a virtual world. Interactive Fiction, or IF, is an example of a kind of interactive fiction in which the computer creates a text version of the tale.

3D simulated worlds (similar to video games) can be accessed via the user's written commands (where the story generation module is used to drive the behavior of virtual characters and the story is only rendered visually). Too many interactive storytelling applications to include in this review, but they clearly merit a separate investigation and review effort.

1. Narratological notions also play an important role in the development of these applications, which form an important study area that brings together computing and narratology.
2. As a whole, this group of study lines is being referred to as Computational Narratology (Computational Narratology), which has lately seen a major expansion over the last several years.

# Chapter 11; Discussions and Further Evaluation

Besides traditional storytellers, there has been a massive increase in interactive storytelling apps. As the name suggests, these are interactive computer programs that let the user to control the actions of an individual character in a virtual world. Interactive Fiction, or IF, is an example of a kind of interactive fiction in which the computer creates a text version of the tale.

3D simulated worlds (similar to video games) can be accessed via the user's written commands (where the story generation module is used to drive the behavior of virtual characters and the story is only rendered visually). Too many interactive storytelling applications to include in this review, but they clearly merit a separate investigation and review effort.

## 11.1 Other Examples of Applications

Systems that create narrative text from a theoretically represented narrative discourse fall under the same general category. Just like in STORYBOOK, by Charles Callaway and James Lester (2002). In contrast, the nn system's Narrator module, developed by Nick Montfort, uses underlying fabula to create tale discourse. It is currently known as the Curveship system (Montfort 2011), or the Suspenser (Cheong 2007) and Prevoyant (Bae & Young 2008) systems, which attempt to produce discourses with certain qualities like suspense and surprise. Algorithms for activities that are plainly crucial in the production of tales include all of these categories as examples (Vincent, 2019).

1. Although they are related to the other systems discussed above, they have been treated separately since their goal is not to generate a narrative but rather to convey it in a certain manner.
2. A similar set of programs seeks to create a cinematic visual narrative from a fictitious premise (Jhala & Young 2010).
3. Narratological notions also play an important role in the development of these applications, which form an important study area that brings together computing and narratology.

As a whole, this group of study lines is being referred to as Computational Narratology

(Computational Narratology), which has lately seen a major expansion over the last several years.

## 11.2 Narratology Relevance

An early prototype of a story generation system may plainly be seen here. In their present condition, they are clearly incapable of providing the depth and complexity of story that narratology is most concerned with. Narratology, on the other hand, has not paid much attention to these concerns, although they might be extremely beneficial if they were included in the design and construction of these systems (James, 2019). The most important aspect of this study from a narratological perspective is the "testing" of narratological ideas for their clarity and application, which are essential for an algorithm-driven system to work (Pablo, 2020).

A wide range of ideas is required for literary analysis, but a narrower range is required for the creation of algorithm-driven systems, which are built on exact specifications for narrative building choices. Prototypes of narrative ideas may be identified by using storytelling systems to facilitate story-building choices (Pablo, 2020). Both existing ideas that are unclear or ambiguous in their current formulations and new concepts related to the creation of tales might benefit from this approach, which can identify and propose new concepts that may be worthy of future study (James, 2019).

# Chapter 12; Future Investigation

The development of narrative generation algorithms is a constant endeavor. To yet, the connection between artificial intelligence and narratology on this topic has been limited. Computational narratology, on the other hand, is showing signs of growth. This is an uncharted territory that should be investigated more, since each of these areas has the potential to greatly contribute to the others. One of the most important things to keep in mind while developing an algorithm for narrative production is to make sure that the entire effort is broken down into smaller, more manageable jobs.

For this to work it may be necessary to make a separation between the creation of a tale and the delivery of that story after it has been created. As you would expect, these two procedures are linked together in fiction, but their inputs and outputs are fundamentally distinct. Narrative theory may make a substantial contribution to storytelling if it could elucidate the relationships and interactions between these two processes. Research into the production of stories has the potential to offer a useful standard for evaluating how well narratological notions may be translated into storytellers' actual functioning implementations.

# Chapter 13; Conclusion

Detailed descriptions and analyses of the algorithms used to construct the levels in this research provided. We will employ a variety of algorithms, including those based on data, case study analysis, and roles played by participants. Prior to implementing any algorithms for story generation, a comprehensive literature study conducted. The current crop of tale generators are limited in their utility since they do not allow for human input.

I started with using GPT2 basics and then learned about all the kinds of searching and sampling used for this. Also took the help of this model to understand all the aspects of Story generation and its different applications, I also tried to solve different use cases we have like giving answers, generating different types of stories, and all. End we also implemented a Story generation and Narration engine based on this model to get and narrate the story.

Every time a tale advances ahead in time, a set of questions are offered to the user and he or she must answer them to continue the story. It will be able to modify the tale's flow depending on the responses supplied by the story generator, such as which questions to ask next and how the story proceeds. This user-driven approach to narrative development examined in further detail in the sections that follow.

# 

# Chapter 14; References

[1]. ook, William Wallace (2020). Plotto. Battle Creek, MI: Ellis Publishing Company. p.

314.

[2]. Eger, Markus, Colin M. Potts, Camille Barot, and R. Michael Young. 2015. “Plotter:

Operationalizing the Master Book of All Plots.” In Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference, 30–33. Association for the Advancement of Artificial Intelligence.

[3]. Hogan, Robert J. (October 2021). "Inside an Author's Brain: The Birth of a Book Length Mystery Novelette". Writer's Digest.

[4]. van Hise, James (2021). Pulp Masters. pp. 120–125.

[5]. Bringsjord, Selmer; Ferrucci, David A. (2020). Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine. Mahwah, New Jersey: Lawrence Erlbaum Associates. pp. 122, 150. ISBN 0-8058-1986-X.

[6]. Meehan, James (2021). "TALE-SPIN". In Schank, Roger C.; Riesbeck, Christopher K. (eds.). Inside Computer Understanding: Five Programs Plus Miniatures. New Haven, Connecticut: Lawrence Erlbaum Associates. p. 197. ISBN 0-89859-071-X.

[7]. Pérez y Pérez, Rafael (2019). "From MEXICA to MEXICA-Impro: The Evolution of a Computer Model for Plot Generation". In Besold, Tarek Richard; Schorlemmer, Marco; Smaill, Alan (eds.). Computational Creativity Research: Towards Creative Machines. Atlantis Press.

[8]. Lönneker, Birte; Meister, Jan Christoph; Gervás, Pablo; Peinado, Federico; Mateas,

Michael (June 2020). Story Generators: Models and Approaches for the Generation of Literary Artefacts. Joint Conference of the Association for Computers and the Humanities/Association for Literary and Linguistic Computing. University of Victoria.

[9]. Gervás, Pablo. "Story Generator Algorithms". The living book of narratology. Retrieved 12 October 2017.

[10]. Vincent, James (14 February 2019). "OpenAI's new multitalented AI writes, translates, and slanders". The Verge. Retrieved 23 February 2021.

[11]. Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J. and Liu, R., 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

[12]. Zhang, H., Song, H., Li, S., Zhou, M. and Song, D., 2022. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*.

[13]. Tan, B., Yang, Z., AI-Shedivat, M., Xing, E.P. and Hu, Z., 2020. Progressive generation of long text with pretrained language models. *arXiv preprint arXiv:2006.15720*.

[14]. Dale, R., 2021. GPT-3: What’s it good for?. *Natural Language Engineering*, *27*(1), pp.113-118.

[15]. "AI Storytelling Game May Expand Publishing's Horizons". www.publishersweekly.com. 28 August 2020. Retrieved 23 February 2021.

[16]. [Pellowski, Anne](https://en.wikipedia.org/wiki/Anne_Pellowski) (2018). [The World of Storytelling.](https://books.google.com/books?id=hDDaAAAAMAAJ) H.W. Wilson (published 1990). p. 44. [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [978-0-8242-0788-5.](https://en.wikipedia.org/wiki/Special:BookSources/978-0-8242-0788-5) Retrieved 27 March 2021. Religious storytelling is that storytelling used by official or semi-official functionaries, leaders, and teachers of a religious group to explain or promulgate their religion through stories

[...].

[17]. Birch, Carol and Melissa Heckler (Eds.) 2019. Who Says?: Essays on Pivotal Issues in Contemporary Storytelling Atlanta GA: August House

[18]. Ruediger Drischel, [Anthology Storytelling](http://anthologystorytelling.com/) - Storytelling in the Age of the Internet, New Technologies, Artificial Intelligence. Retrieved January 15, 2019

*[19].* Bae, Byung-Chull, & R. Michael Young (2008). “A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach.” U. Sperling & N. Szillas (eds). *Interactive Storytelling. First Joint International Conference on Digital Storytelling, ICIDS 2008*, Erfurt, Germany, 26–29. Proceedings Series: Lecture Notes in Computer Science, Vol. 5334. Berlin: Springer, 156–67.

[20]. Bringsjord, Selmer, & David A. Ferrucci (1999). *Artificial Intelligence and Literary Creativity Inside the Mind of BRUTUS, a Storytelling Machine*. Hillsdale, N.J.: Lawrence Erlbaum.

[21]. Callaway, Charles B., & James C. Lester (2002). “Narrative prose generation.” *Artificial Intelligence* 139.2: 213–52.

[22]. Cheong, Yun-Gyung (2007). *A Computational Model of Narrative Generation for Suspense*. PhD Thesis, North Carolina State University.

[23]. Dehn, Natalie (1981). “Story Generation after Tale-Spin.” A. Drinan (ed). *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, August 24–28, University of British Columbia, Vancouver, [24]. Canada. Los Altos, CA: Kaufmann, vol. 116–18.

[25]. Gervás, Pablo (2009). “Computational approaches to storytelling and creativity.” *AI Magazine* 30.3: 49–62.

[26]. Jhala, Arnav, & R. Michael Young (2010). “Cinematic visual discourse: Representation, generation, and evaluation.” *IEEE Transactions on Computational Intelligence and AI in Games* 2.2: 69–81.

[27]. Klein, Sheldon, et al. (1973). *Automatic novel writing: A status report. Technical Report* 186, Computer Science Department, The University of Wisconsin, Madison.

[28]. Lebowitz, Michael (1983). “Creating a Story-Telling Universe.” A. Nundy

(ed). *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, August 8–12, Karlsruhe, Germany. Los Altos, CA: Kaufmann, vol. 1, 63–

65.

[29]. Lönneker, Birte, et al. (2005). “Story Generators: Models and Approaches for the Generation of Literary Artefacts.” *ACH/ALLC 2005 Conference Abstracts. Proceedings of the 17th Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, Victoria, BC, Canada, June 15–18, 126–33.

[30]. Mani, Inderjeet (2010). *The Imagined Moment: Time, Narrative, and Computation*. Lincoln: U of Nebraska P.

[31]. Meehan, James R. (1977). “Tale-Spin, an interactive program that writes stories.” *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, MIT, Cambridge, MA, August 22–25. Los Altos, CA: Kaufmann, 91–98.

[32]. Meister, Jan Chistoph (2005). “Computational approaches to narrative.” D. Herman et al. (eds). *Routledge Encyclopedia of Narratology*. London: Routledge, 78–80.

[33]. Montfort, Nick (2007). *Generating narrative variation in interactive fiction*. PhD Dissertation, University of Pennsylvania, Philadelphia.

[34]. Montfort, Nick (2011). “Curveship’s automatic narrative variation.” *Proceedings of the 6th International Conference on the Foundations of Digital Games* (FDG ’11), Bordeaux, France, June 29–July 1. New York: ACM, 211–18.

[35]. Pérez y Pérez, Rafael (1999). *MEXICA: A Computer Model of Creativity in Writing*. PhD Dissertation, The University of Sussex.

[36]. Riedl, Mark O., & R. Michael Young (2010). “Narrative Planning: Balancing Plot and Character.” *Journal of Artificial Intelligence Research* 39: 217–68.

[37]. Turner, Scott R. (1993). *Minstrel: a computer model of creativity and storytelling*. PhD Dissertation, University of California at Los Angeles, Los Angeles.

[38]. Bailey, Paul (1999). “Searching for storiness: Story generation from a reader’s perspective.” *Narrative Intelligence. Papers from the 1999 AAAI Fall Symposium*.

North Falmouth, MA, November 5–7. Menlo Park, CA: AAAI Press, 157–63.

[39]. Lebowitz, Michael (1985). “Story-telling as planning and learning.” *Poetics* 14: 483–502.

[40]. Meehan, James R. (1981). “Tale-Spin.” R. Schank (ed). *Inside Computer Understanding*. Hillsdale, NJ: Lawrence Erlbaum, 197–225.

[41]. Pérez y Pérez, Rafael, & Mike Sharples (2004). “Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA.” *Knowledge-Based Systems* 17:

15–29.

[42]. Riedl, Mark O., & Neha Sugandh (2008). “Story planning with vignettes: Toward overcoming the content production bottleneck.” U. Sperling & N. Szillas (eds). *Interactive Storytelling. First Joint International Conference on Digital Storytelling, ICIDS 2008*, Erfurt, Germany, 26–29. Proceedings Series: Lecture Notes in Computer Science, Vol. 5334. Berlin: Springer, 168–79.

[43]. Turner, Scott R. (1994). *The Creative Process: A Computer Model of Storytelling and Creativity*. Hillsdale, NJ: Lawrence Erlbaum.