DEVELOPING AN EVALUATION FRAMEWORK FOR AI-GENERATED SCREENPLAYS

A Thesis

Presented to the

Faculty of

California State Polytechnic University, Pomona

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

In

Computer Science

By

Zhong Ooi

2024

COMMITTEE MEMBERSHIP

THESIS: DEVELOPING AN EVALUATION FRAMEWORK FOR AI-GENERATED SCREENPLAYS

AUTHOR: Zhong Ooi

DATE SUBMITTED: Fall 2024

Department of Computer Science

Dr. Yu Sun
Thesis Committee Chair
Professor of Computer Science

Dr. Markus Eger
Professor of Computer Science
UC Santa Cruz, Department of Computational Media

Dr. Ben Steichen
Professor of Computer Science

ACKNOWLEDGMENTS

I would like to first express my deepest gratitude to Professor Markus Eger for all the support he provided for this thesis. Through thick and thin, as I feature crept this thesis to hell and back, he provided valuable insight and encouragement through the process. His support was unwavering even after transitioning to a new institution, was unprecedented. Professor Eger helped me find a new professor during my transition period and provided consistent guidance through numerous online meetings throughout the year. I am particularly grateful for his support in funding the generation of numerous GPT-4 Turbo screenplays, which was pivotal to the depth and scope of this paper. Without his dedication and assistance, this work would only be a fraction of what it is today.

I would also like to sincerely thank Professor Yu Sun and Ben Steichen for their swift help during the transition period. When my Thesis Committee Chair moved to another institution, I was initially overwhelmed by the challenge of this sudden change. However, Professor Sun's timely support and willingness to take on the role of committee chair, along with Professor Ben Steichen's readiness to join the committee, enabled a seamless transition and allowed me to continue my research without undue stress.

To my family, thank you for your unwavering support throughout my college journey. Even during moments when I was reserved and didn't want to share details during this process, your encouragement and reassurance gave me the strength to complete this paper.

To my friends, you made my master's program an unforgettable and joyous experience. This year and a half has been among the best years of my academic life, and I wouldn't trade those memories for anything.

Finally, I thank one specific friend, Jordan Lin, for his technical and steadfast support. Throughout the writing of this paper, he was a constant source of assistance, helping troubleshoot issues with ScreenPy, forced directed graph visualization, SVO identification software, and developing the tables displaying results. His contributions enhanced the visualization's clarity and robustness, allowing more clarity in understanding the results.

ABSTRACT

Evaluating artificial intelligence models is a crucial step in their development, providing insights into performance. While classification and prediction models are often assessed using quantitative metrics, evaluating text-based generative models presents unique challenges. These models produce novel and unpredictable outputs, making rigid quantitative metrics insufficient to capture the generated content's creativity and nuance.

Screenplays are characterized by a well-defined structure, which makes them ideal for data scraping and analysis. Moreover, screenplays are readily available, providing a diverse dataset that can be used as a benchmark to gauge the quality and cohesiveness of AI-generated narratives. By benchmarking AI-generated content against human screenplays, this program aims to provide an evaluation framework that incorporates a variety of qualitative measures to test the performance of these AI generation models, such as syntactic complexity, sentiment analysis, part of speech analysis, and character presence visualizations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1:

Introduction

In recent years, there has been a substantial rise in the capabilities of generative artificial intelligence models. One of the most notable risers has been OpenAI GPT models[1]. The GPT models are transformer-based natural language processing models designed to generate, classify, and summarize text with high levels of coherence and accuracy [1]. It accomplishes this through a probabilistic approach, leveraging advanced methods to predict the most likely next word in a sequence based on a given instruction or prompt. Despite their advanced capabilities, testing generative models poses an inherent challenge. These models are designed to create novel and unique generative text, making it inherently difficult to compare their outputs to a predefined ground truth.

For smaller generative tasks, such as generating individual sentences, the output size is manageable enough for human evaluation. Additionally, there are established benchmarking tools, like the GLUE (General Language Understanding Evaluation) benchmark, that provide standardized testing methods to evaluate language models on tasks involving smaller text units, such as acceptability, sentiment analysis, paraphrase, sentence similarity, natural language inference, and question answering [2]. GLUE evaluates models by comparing their performance on these specific, predefined language understanding tasks, offering a measure of their capabilities when dealing with concise outputs. However, for more complex, longer-form generative tasks—such as generating entire screenplays, stories, or articles—evaluating the quality of the output becomes significantly more challenging. The lack of a direct, easily comparable ground truth makes it difficult to establish objective measures of quality or coherence. Human evaluation for such large outputs is often unreasonable as the size of this generation would be subjective and resource-intensive to comb through.

For long-form generation, screenplays are exceptionally well-suited for analysis due to their

[1] https://openai.com/

structured and standardized format, which offers a clear framework for data scraping and analysis. Hoyt, Ponto, and Roy highlight this in their work, stating that "the motion picture screenplay may be the most perfectly pre-disposed for computational analysis...As semi-structured documents with formatting conventions analogous to a metadata schema, screenplays are ideally suited for automated computer parsing. There is no need for laborious TEI encoding to detect character dialogue exchanges and interactions." [3] This structure makes it easier to apply analysis techniques to scrape screenplay data and understand narrative flow, character interactions, and other key storytelling elements. In addition to being predisposed to computational analysis, a wealth of screenplays is readily available online through the Internet Movie Script Database (IMSDb)[2]. Furthermore, the accessibility of screenplays online, through repositories like the Internet Movie Script Database, enhances their availability for research. IMSDb hosts an extensive collection of screenplays spanning various genres and time periods, making it an invaluable resource for obtaining a diverse dataset. By parsing these screenplays and applying evaluation software, researchers can establish benchmarks based on human-authored examples. This enables a robust comparison of AI-generated screenplays against human-written counterparts, providing critical insights into the quality and creativity of automated narrative generation.

When analyzing longer text such as screenplays, there are two options: close reading or distant reading [4]. Close reading is an analytical method with a more granular focus, allowing for retention of the source text. It emphasizes detailed examination to uncover the text's complex meanings and subtle themes. Distant reading is an analytical method where the focus is on a more holistic level, losing retention of the source text and instead generating an abstract view of the whole text. It involves analyzing text using computational techniques where distant reading examines patterns, trends, and relationships across the full texts instead of focusing on individual sections. It emphasizes understanding a collection's broader context and overall structure, which would be impossible to analyze manually.

[2]https://imsdb.com/

Since long-form content is inherently difficult to evaluate due to its complexity, scale, and the subjective nature of storytelling, I implemented a systematic evaluation framework to address these challenges. In this thesis, I explored the development of a comprehensive screenplay evaluator designed to assess, analyze, and compare screenplays' structural, stylistic, and narrative elements. The primary goal was to create a tool capable of evaluating screenplays from human-authored and AI-generated, providing meaningful insights into these AI models' qualitative and quantitative characteristics. By focusing on a distant reading approach, the evaluator evaluates the screenplay over trends rather than individual use of language, allowing for a systematic and abstract examination of narrative flow, linguistic complexity, and structural coherence. The evaluator examines multiple facets of the screenplay through evaluation methods such as presence graph, increasing graph, heaps law, scene length by sentence and words, part of speech, Gini index, Yngves, and Frazier mean complexity score, sentiment analysis, chord graph, forced directed graph, and subject-verb-object triplets.

Related Work

The computational analysis of screenplays is not a novel area of exploration, with notable contributions emerging from the digital humanities. Several programs have been developed to scrape screenplays and produce visualizations that assist in analyzing structural and narrative elements. For instance, Scripthreads, developed by Hoyt, Ponto, and Roy, generates multiple visualizations to analyze aspects such as scenes, pacing, and character interactions [3]. Among the visualizations produced are the increasing graph, presence graph, and force-directed graph, all of which were partially emulated in this thesis.

While Scripthreads demonstrates significant utility, it falls short in addressing the suite of evaluations required for this work. One major limitation of the system was its rigidity. Distributed as a standalone executable, it lacked extensibility, making it challenging to integrate additional evaluation metrics or customize its functionality for diverse analytical needs. Another significant drawback was its inability to parse the screenplay in a timely manner. Extensive testing revealed

that the parser used in the system failed to capture all characters in a screenplay, necessitating a labor-intensive manual correction process. While this issue might not be critical for analyzing individual screenplays, it posed a considerable challenge for our purposes, as working with hundreds of screenplays requires a more automated solution.

Other tools designed for screenplay analysis, such as Screenplay Owl, developed by Samuel Marinov and Brock Stitts and ScriptFAQ, developed by Stewart McKie, have similarly contributed to this field.[3]Unfortunately, many of these tools are no longer accessible, further underscoring the need for a robust, extensible framework for screenplay analysis that can evolve alongside advancements in computational methods and user needs.

Beyond screenplay-specific analyzers, research in text analysis more broadly has provided valuable insights for understanding narrative structures. Roberto Franzosi's work in Quantitative Narrative Analysis contributes significantly to narrative analysis by applying a big data approach to explore and quantify elements of creative texts [5]. Franzosi emphasizes the value of using quantitative methods to interpret the underlying dynamics of storytelling, facilitating a data-driven understanding of literary constructs through techniques like sentiment analysis, sentence complexity, and part-of-speech analysis.

Similarly, the work of Saatviga Sudhahar and Nello Cristianini demonstrates how text mining techniques can be used for automatic subject-verb-object extraction[6]. The resulting subject-verb-object triplets provide a robust, succinct representation of a written text and have been used to generate visualizations such as network maps, subject/object bias analysis, and verb clouds.

In addition to research focused on text analysis, significant work has been done on generating long-form content. For instance, Maryam Dueifi and Markus Eger explored how a Glaive planner could be utilized to guide the generation of novella-length stories. Their work highlights the potential of structured planning to enhance extended texts' coherence and narrative quality. Given the parallels in generating long-form content, my work draws foundational insights from their approach to setting up API calls for story generation [7].

Background

According to Rex Provost, "A screenplay is a written work for a film, television show, or other moving media, that expresses the movement, actions and dialogue of characters. Screenplays, or scripts, are the blueprint for the movie. "[8] Like any blueprint, the modern screenplay adheres to specific formatting guidelines, including standardized font, page margins, page numbering, and length. It also sets guidelines for distinct textual elements such as scene headings, action lines, character names, dialogue, and parentheticals. For a distant reading approach to screenplay evaluation, the most critical elements to analyze are the textual elements, such as scene headings, action lines, and character dialogue. These components form the backbone of a screenplay's structure and narrative flow, making them essential for analysis. To clarify what these elements represent and how they function within a screenplay, I will describe each using an excerpt from the opening scenes of Indiana Jones and the Raiders of the Lost Ark, as illustrated in Fig.1.1.

EXT. PERU - HIGH JUNGLE - DAY

The dense, lush rain forests of the eastern slopes of the Andes, the place known as "The Eyebrow of the Jungle". Ragged, jutting canyon walls are half-hidden by the thick mists.

The MAIN TITLE is followed by this:

PERU
1936

A narrow trail across the green face of the canyon. A group of men make their way along it. At the head of the party is an American, INDIANA JONES. He wears a short leather jacket, a flapped holster, and a brimmed felt hat with a weird feather stuck in the band. Behind him come two Spanish Peruvians, SATIPO and BARRANCA. Bringing up the rear are five Yagua INDIANS. They act as porters and are wrangling the two heavily-packed llamas. The Indians become increasingly nervous. They speak to each other in bursts of Quechua. The American, who is known to his friends as Indy, glances back at them.

BARRANCA
(irritated)
They're talking about the Curse again!

Figure 1.1: Excerpt from the first scene of *Indiana Jones Raider of the Lost Ark*

The scene heading specifies the time and location where a scene takes place [8]. For example, the scene heading in Fig. 1.1 is written as **EXT. PERU - HIGH JUNGLE - DAY**. To indicate whether the scene is indoors or outdoors, the terms **INT.** (interior) and **EXT.** (exterior) are commonly used. Additionally, the hybrid **INT./EXT.** can denote scenes that transition indoors and outdoors or when the distinction is unclear. In Fig. 1.1, the **EXT.** indicates that the scene takes place outdoors. The second portion, **PERU**, denotes the general location. For more specific details, additional descriptors can be included; in this case, **HIGH JUNGLE** provides a more precise description of the setting. Finally, The time of day is typically indicated at the end of the scene heading. Common terms include **Morning**, **Night**, **Continuous**, or **Later**. In the example from Fig. 1.1, the scene occurs during the **Day**.

Action lines, written in the third-person point of view and present tense, are where all non-dialogue elements of the screenplay are conveyed [8]. These include scene descriptions, character introductions, and actions. While action lines encompass everything outside dialogue, they also adhere to specific conventions. The characters' names the first time they appear, as well as sound effects, key details, props, and specific shots, will be written in uppercase. These conventions ensure clarity and emphasize key elements of the screenplay. The action lines in Fig. 1.1 include all the text between the scene heading and Barranca's dialogue. As shown in the excerpt, these lines provide a vivid description of the scenery, details about Indiana Jones' appearance, and an overview of the group's actions additionally, since this is the first time **BARRANCA**, **SATIPO**, and **INDIANA JONES** are introduced, their names are written in uppercase, adhering to screenplay conventions for highlighting new characters.

The character section, which includes the character's name and dialogue, indicates the lines spoken by each character [8]. The character cue, written in uppercase and centered above the dialogue, clearly identifies who is speaking. In some cases, special notations such as **(V.O.)** for "voice over" or **(O.S.)** for "off-screen" appear next to the character's name to clarify the context in which the dialogue occurs. Additionally, parentheticals may be included beneath the character's

name to provide brief descriptions or instructions regarding tone, emotion, or specific actions while delivering the lines, ensuring the dialogue is executed as intended. In the excerpt from Fig. 1.1, the character cue **BARRANCA** is centered above the dialogue to indicate who is speaking. Below the cue, the parenthetical **(irritated)** describes the tone in which the line is delivered. Finally, the dialogue, *"They're talking about the Curse again!"*, appears directly beneath the character cue and parenthetical, representing what **BARRANCA** says in the scene.

Chapter 2:

Methodology

Screenplays present significant challenges for evaluation due to their scale and the subjective nature of storytelling. To address these challenges, I created a comprehensive screenplay evaluator designed to analyze and compare screenplays' structural, stylistic, and narrative elements. The primary aim was to build a tool capable of evaluating human-authored and AI-generated screenplays, offering meaningful insights into these AI models' qualitative and quantitative characteristics. Evaluating generated screenplays involves three key stages: screenplay generation, information extraction, and comprehensive evaluation. In this section of the paper, I delve into each stage to provide a clearer understanding of the workflow involved.

Screenplay Generation

When using the GPT API, it is crucial to understand the roles of prompts, conversation styles, roles, and temperature settings, as these parameters can influence the generation of screenplays. These factors impact the output in various ways, such as the variability in language, the coherence and interconnectivity of characters and locations, the overall quality of the narrative, and adherence to proper screenplay formatting. For this thesis, I primarily utilized the GPT-3.5-turbo and GPT-4.0-turbo models, leveraging their advanced capabilities to explore and refine the process of screenplay generation.

Prompting

Before generating any output, crafting a prompt that effectively guides the system to produce the desired results is crucial. This is particularly important in automated generation, as an ineffective prompt can lead to errors in output formatting, making subsequent processing more challenging. When setting up prompts for OpenAI API calls, three roles help guide the generation process: system, user, and assistant.

System Role

The system role provides top-level instructions to the model, outlining its intended function and guiding its general behavior and response style.[1]. This role is crucial in establishing the foundation for the formatting and rules that the generated output must follow. It ensures adherence to the highly structured format of screenplays while maintaining consistency and predictability in formatting, which is essential for efficient data scraping and extraction. To guide the generation, the prompt below is used in every API call of the screenplay.

```
"You are a screenwriter's assistant.
Format the screenplay so it would be in the left justified of the page.
Your task is to generate screenplay scenes that include the following:
1. Scene heading: INT./EXT. LOCATION - TIME OF DAY.
2. Action: Use descriptive paragraphs to set up the scene, location, and
environment. Use short and visual description to set up the scene
and environment.
3. Character: The name of the person speaking, in all caps.
4. Dialogue: What the character says.
5. Parenthetical: If necessary, to describe how the dialogue is delivered."
```

This prompt helps guide the generation of screenplays by specifying and concisely explaining several critical elements of the screenplay, such as the scene heading, action lines, character cues, dialogue lines, and parenthetical[8]. Using these screenplay conventions helps ensure a structured and professional format that adheres to industry standards. Aside from the elements of the screenplay, the formatting for data extraction must also be present in the prompt. Specifying, ”Format the screenplay so it would be in the left-justified of the page.”, though inconsistent with screenwriting formatting standards, helps ensure that the output follows a predictable and uniform pattern for

[1] https://platform.openai.com/docs/guides/text-generation

easier data analysis.

User Role

Beyond crafting prompts for formatting and detailing elements of a screenplay, it is equally important to develop user prompts. The user role represents instructions that request a particular type of output from the model[2]. The user prompt for our purposes serves as the blueprint for shaping the story scene by scene, providing direction for the AI's generation. The prompt can be written in such a manner to define key elements such as story beats, interpersonal conflicts, location details, or specific plot points that can be incorporated into each scene. For example, with the prompt "Write the beginning scene of a screenplay on a tragedy between two star-crossed lovers who will suffer and die. Their family will also kill each other on sight." created a short screenplay that used elements and characters from the popular story "Romeo and Juliet" as seen in Fig. A.1.

However, for this thesis, it is more important to see the baseline of the model's generation rather than a possible rehashing of an existing story. The prompts used to generate scenes were intentionally designed with minimal information to minimize potential bias. This approach reduces the likelihood of the model overemphasizing words from the prompt or closely adhering to existing stories, ensuring a more independent and original generation process. Following the prompts used by Dueifi and Eger, but omitting instructions for planner direction, the model is given only three elements from the user prompt: the genre, the current scene count, and the previous scene(s)[7]. Each piece of information gives vital direction without adding too much influence. The genre helps align the generated screenplay with genre-specific conventions and expectations, making it easier to compare to human-authored screenplays. The chapter count guides the story's progression and the amount of text generated, helping the model determine the appropriate pacing and structure relative to where it is in the narrative. By providing the previous scene, the model can build upon the story already generated.

[2]https://platform.openai.com/docs/guides/text-generation

Conversation Type

When using GPT, there are two distinct approaches to interacting with previous scenes for content generation: single-turn and multi-turn conversations. By generating screenplays using these two conversation approaches, the aim was to understand how these approaches impact the quality, coherence, and creative flow of screenplay generation. Each generation for both methods involved generating 50 scenes of a screenplay in the adventure genre, allowing us to compare how each conversation style handles long-form storytelling and whether consistent quality could be maintained across a substantial number of scenes.

The single-turn conversation approach generates each section of content independently, without leveraging any memory or persistent contextual awareness from the model [7]. This method provides context solely through the prompt, which includes the previous scene. Although the model receives information about recent developments, such as events or character actions, its understanding is limited to what is immediately provided in the input prompt, without any cumulative learning or recall across multiple turns.

The multi-turn conversation approach generates each section of content while preserving memory and maintaining contextual carry-over from the model. This method provides context through the assistant's ability to recall previously generated scenes. This allows the model to build on established characters, themes, and plotlines as the narrative unfolds, ensuring greater continuity. Managing the model's memory poses a significant challenge because the token limit constrains the total amount of text the model can handle in a conversation. If this issue is not addressed, content generation risks producing a brief story or failing once the token limit is exceeded. To overcome this, I implemented a strategy in which the assistant retains memory of only the most recent five scenes, gradually removing older scenes from the message history as new content is generated. This sliding window approach ensures the model remains focused on recent developments, preserving crucial context while effectively managing the token limit.

Model Temperature

In OpenAI models, the temperature setting is a parameter that controls the randomness or creativity of the outputs[3]. In other words, temperature influences the deterministic or diverseness of the generated responses. A lower temperature produces more deterministic and predictable responses, whereas a higher temperature encourages a more stochastic approach, adding variability to the generated content. In this thesis, I used temperature values of 0.2, 0.6, and 1 to determine how big of an effect the temperature had on the quality and characteristics of the generated content. By testing these different temperature settings, I sought to understand how varying degrees of randomness affect the overall quality and cohesion of the generated screenplay. I comprehensively assess how these parameters influenced the generative process by combining these different temperature settings with our two conversation styles. This exhaustive approach could reveal which combinations struck the most compelling balance between creativity and coherence, providing valuable insights for optimizing screenplay generation.

Information Extraction

Information extraction is a crucial component in screenplay evaluation, serving as a foundational process for identifying and extracting narrative elements. This procedure is instrumental in breaking down a screenplay into its fundamental components of action lines and character dialogue, allowing for further analysis from evaluators. The information extraction process can be divided into two essential steps: parsing and data formatting.

Parsing

Screenplays typically adhere to a standardized format, allowing systematic isolation and parsing of textual elements such as scene headings, action lines, and character dialogue. However, while most screenplays generally follow these conventions, each writer often introduces subtle variations to the formatting, adding slight inconsistencies in the raw text. Consequently, an information extraction program must be adaptable enough to account for these variations. This flexibility is

[3] https://platform.openai.com/docs/guides/text-generation

particularly important when working with IMSDb, the largest screenplay database, which provides only raw text without accompanying metadata. As a result, effective parser is essential to extract meaningful information from the screenplays.

Human-Authored Screenplay Parsing

For human-authored screenplays that adhere to traditional screenplay formatting conventions, I utilized David R. Winer's ScreenPy program[4] to parse and extract data from each screenplay[9]. ScreenPy is a recursive descent parser specifically developed to handle the structured format of screenplays by efficiently breaking down and categorizing the various elements within the text. ScreenPy first decomposes the screenplay into scenes and then breaks each into distinct segments, identified as stage directions(action lines) or dialogue. When a scene heading is identified, it is anchored to the first stage direction in which relevant information appears. This scene is then divided into key components: subject, location, interior/exterior, time of day (TOD), and shot type. Winer organizes the screenplay as an ordered list of scenes, each consisting of an ordered list of segments. This hierarchical structure makes the parsed data easy to analyze, ensuring that each scene and its components are clearly separated and accurately categorized for subsequent processing.

The dictionary structure used to store action lines and dialogue lines is illustrated in A.1 for action lines and A.2 for character dialogue. These two dictionaries share the same three distinct keys designed to specify and organize the parsed portions of the screenplay text.

- head_type: This key identifies the type of each line or element in the screenplay. For example, it can specify whether the line is a stage direction or a dialogue line.
- head_text: The content of this key varies depending on the value of head_type. For scene headings, head_text contains the scene heading, broken down into its components (e.g., interior/exterior indicator, location, and time of day). For character names or titles, head_text stores the name of the character speaking or the title associated with the section.

[4]https://github.com/drwiner/ScreenPy/tree/master

- text: This key stores the actual text found at the location specified by the other keys. For example, in the case of dialog, it would store the spoken lines, while for scene descriptions, it would contain the descriptive text.

Generated Screenplays Parsing

Screenplays generated by ChatGPT often exhibit formatting inconsistencies, particularly with spacing intended to center the text on a page. These irregularities in the generated formatting posed challenges for ScreenPy, making it difficult to parse the screenplay consistently. To address this issue and ensure a more predictable output, I employed system prompts to enforce left-justified formatting, producing screenplays in a consistent and predictable pattern. Since generated screenplays deviate from traditional screenplay formatting, I developed a new parser mainly using regex statements specifically tailored to handle the unique structure of these generated screenplays. This parser only had one constraint: to capture all the same information as the ScreenPY Parser, which are scene heading, stage direction, and dialogue. Due to the two different formats possible for scene heading I used two different regex statements to capture each version to split it into the subject, location, interior/exterior (terior), and time of day (tod) components found in ScreenPY[9]. For Dialogue and Action Lines, I capture this information by focusing solely on Dialogue. In the system prompt, the dialogue was formatted with a character cue followed by an optional parenthetical and the dialogue text. This predictable formatting allowed me to accurately identify and classify dialogue lines, while all other lines were categorized as action lines.

Data Formatting

With the screenplay scraped, I retrieved three major components: the scene heading, action lines, and dialogue. With this information, I can create a data frame in which all evaluators pull data relating to the following columns: sentence_index, type, terior, heading, subheading, ToD, and text. To get a better understanding of the data formatting, Figure 2.1 provides a snapshot of the DataFrame by displaying both the first five rows and the last five rows. Here's what each column represents:

- sentence_index: The cumulative count of sentences throughout the screenplay, including the text in the current row.
- type: Stores the information if the line is an action line, identified as a heading, or dialogue, as the character name.
- terior, heading, subheading, ToD: Extracted from the scene heading, these columns hold the scene information. For example, with the scene heading EXT. PERU - HIGH JUNGLE - DAY, I would store EXT is used for terior, PERU is used for heading, Null is used for the subheading, and DAY is used for ToD.
- text: The raw text from the screenplay can either be action descriptions or dialogue lines following each section.

	sentence_index	type	terior	heading	subheading	ToD	text
0	3	HEADING	EXT.	PERU	['HIGH JUNGLE']	DAY	The dense, lush rain forests of the eastern sl...
1	4	BARRANCA	EXT.	PERU	['HIGH JUNGLE']	DAY	(irritated) They're talking about the Curse ag...
2	13	HEADING	EXT.	PERU	['HIGH JUNGLE']	DAY	He turns and yells at the Indians in Quechua, ...
3	14	INDY	EXT.	PERU	['HIGH JUNGLE']	DAY	No.
4	16	HEADING	EXT.	PERU	['HIGH JUNGLE']	DAY	Barranca looks evilly at Indy's hand upon him....
...	...	...	...	...	...	...	...
849	2549	ARMY INTEL. #9906753	INT.	GOVERNMENT WAREHOUSE			
850	2550	HEADING	INT.	GOVERNMENT WAREHOUSE			DO NOT OPEN! The hammering is completed and ha...
851	2560	HEADING	INT.	GOVERNMENT WAREHOUSE			A Little Old Government Warehouseman begins pu...
852	2560	THE END	INT.	GOVERNMENT WAREHOUSE			
853	2560	HEADING	INT.	GOVERNMENT WAREHOUSE			

Figure 2.1: *Indiana Jones and Raiders of the Lost Ark* DataFrame

With the screenplay stored as a dataframe, I can not only do a full analysis of the screenplay in general but also create subset data frames for deeper analysis in a certain aspect of the screenplay, such as examining character interactions to exploring location usage. The character dataframe allows us to analyze dialogue and the frequency of character appearances specifically. The heading dataframe lets us delve into all portions of a screenplay outside of the character dialogue. Aside

from the dataframe, I also stored the start of each new scene, allowing for scene-by-scene analysis. The location data frame not only helps identify the scene of the different locations by scene but also helps create a character co-occurrence dataframe that reveals patterns of character interactions, highlighting relationships throughout the story. By leveraging these additional dataframes and preserving scene boundaries, I enable individual element analysis and contextual exploration of how these components interact and evolve across the screenplay, providing a richer and more nuanced understanding of its structure and storytelling dynamics.

Evaluator

To effectively evaluate a screenplay both visually and quantitatively, it is crucial to develop a comprehensive suite of evaluators designed to analyze the data collected during the scraping process. These evaluators form the backbone of the assessment framework, enabling a detailed examination of screenplay structure, syntactic patterns, and character presence. The evaluator developed in this thesis incorporates a range of evaluation methods, including presence graphs, increasing graphs, Heap's Law analysis, scene length metrics (by sentences and words), part-of-speech distributions, Gini indices, Yngve's and Frazier mean complexity scores, sentiment analysis, chord graphs, force-directed graphs, and subject-verb-object triplets. This section provides a detailed overview of each evaluator's contributions to the overall evaluation process and how they were constructed. For this section, all the visualizations done are for the screenplay *Indiana Jones and Raiders of the Lost Ark.* This screenplay was chosen due to its popularity in the adventure genre of movies, close following of adventure genre conventions, lack of major errors in the scrapping process, and use in other papers dealing with storytelling[9, 7]. During this section, each character in the screenplay is given their own color, which can be found at A.2

In addition to evaluating screenplays, another key objective is to design the tool to support an extensible framework capable of accommodating diverse analytical methods while maintaining the flexibility to integrate multiple computational techniques and evaluation approaches. To achieve this, each evaluation method is implemented as a subclass of a parent evaluator class, which pro-

vides essential functionality such as reading DataFrames, minimizing the number of input parameters, and generating image filenames. This approach simplifies the development and integration of new evaluators, enabling a modular and scalable architecture. Additionally, a specialized GraphParent class extends the framework by offering advanced features like alternating band visualization and sorting methodologies, further enhancing graph generation and analysis capabilities. Therefore, this design promotes consistency across different evaluation methods and streamlines the workflow, ensuring efficiency and flexibility. Facilitating the seamless addition of new evaluators allows the system to evolve alongside emerging analytical needs and computational methodologies, maintaining its relevance and adaptability. The codebase for the screenplay evaluator developed for this thesis can be found at https://github.com/KnightinGale9/ScreenplayEvaluator.

Presence Graph

The Presence Graph is a visualizes character activity by mapping their appearances in each scene [3]. This visualization offers a comprehensive overview of character involvement and appearance in the story, clearly highlighting when characters are present and when they share screen time with others.

I used an event plot to visually represent the presence graph as seen in Fig. 2.2. In this representation, the x-axis denotes the characters, while the y-axis corresponds to the sentence index, indicating its position within the narrative. Alternating horizontal bands are included to signify transitions between scenes, allowing for a visual differentiator of the scene boundaries. Additionally, characters were sorted based on their co-occurrence sums, grouping characters frequently appearing together in the screenplay for better contextual alignment.

As illustrated in Fig. 2.2, besides identifying the main characters, the graph reveals groups of characters that appear simultaneously, offering insights into narrative structure and relationships. A notable example of this would be the characters Brody, Musgrove, and Eaton, who appear in the same scene together predominantly near the beginning and end of the screenplay.

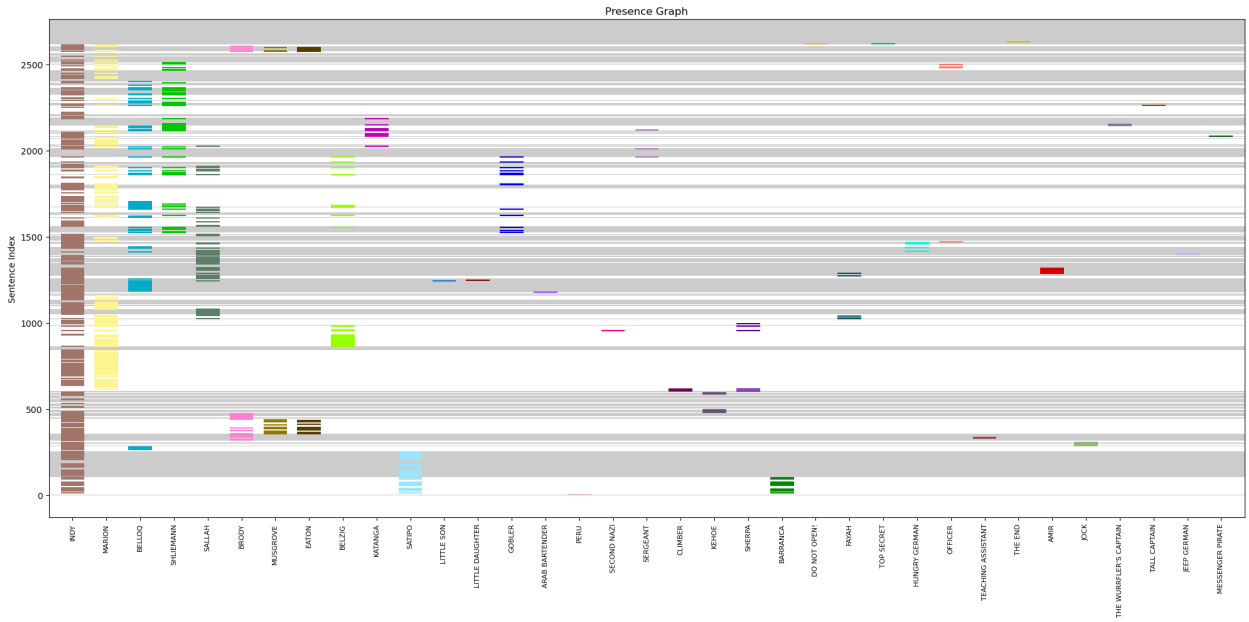


Figure 2.2: Presence Graph for *Indiana Jones and the Raiders of the Lost Ark*

Increasing Graph

The Increasing Graph visualizes character activity aggregated as a line graph across an entire screenplay, focusing on cumulative appearances rather than individual scenes[3]. This method provides a holistic view of character prominence. It allows for a quick assessment of which characters play the most significant roles in the narrative and when the characters become more prominent.

I used a step plot to visually represent the presence graph, as seen in Fig. 2.3. In this representation, the x-axis corresponds to the sentence index, indicating its position within the narrative, while the y-axis corresponds to the character's presence. Alternating vertical bands are included to signify transitions between scenes, allowing for a visual differentiator of the scene boundaries.

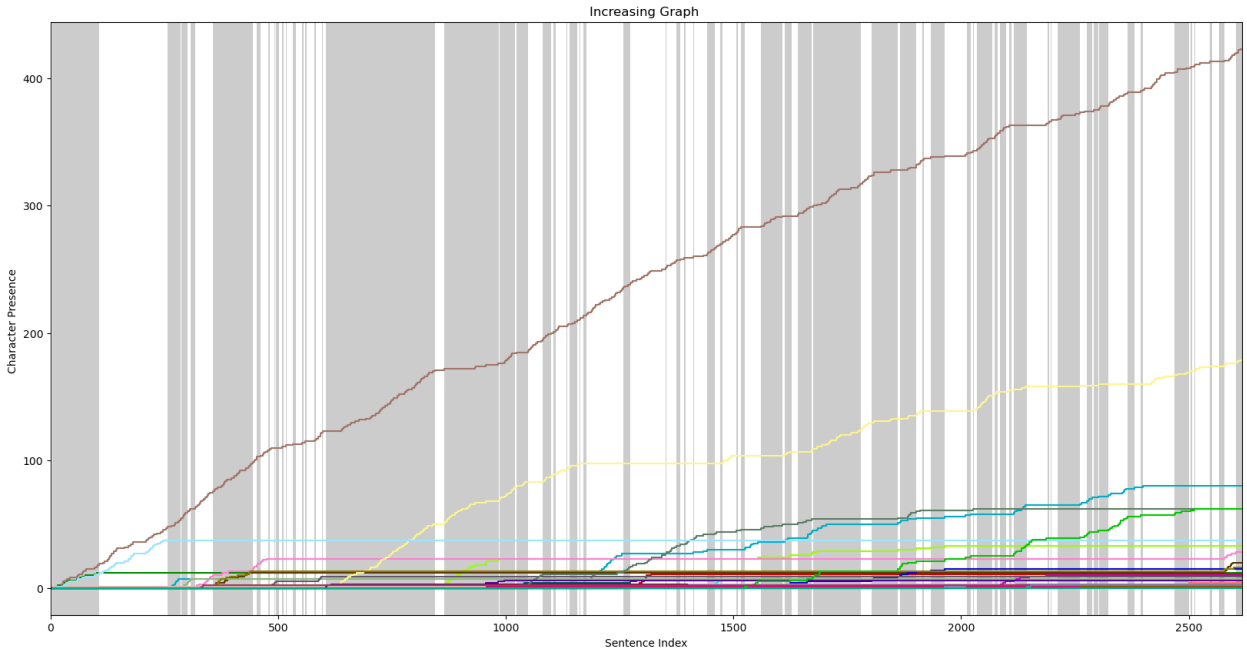


Figure 2.3: Increasing graph for *Indiana Jones and the Raiders of the Lost Ark*

As illustrated in Fig. 2.3, this visualization highlights the main characters of the screenplay: Indy, the protagonist; Marion, the female lead; and Belloq, the antagonist. The prominent presence of Indy compared to all other characters aligns with the "hyper-present protagonist" pattern described in [3]. While this observation may be less critical for analyzing a well-known screenplay, it provides valuable information for generated screenplays, offering a quick insight into which characters are central to the story.

Heaps Law

Heaps Law is a linguistic principle that describes the relationship between the size of a corpus and the number of unique words within the corpus. [10]This analysis offers insight into the depth of the vocabulary, revealing whether the narrative heavily reuses a core set of terms or continually introduces new ones. Additionally, it can identify key points where vocabulary growth accelerates, potentially signaling shifts in setting or character interactions. By graphing heaps of law, I can visually understand the author's use of vocabulary in the screenplay. I used a line graph to visually

represent Heaps Law, as seen in Fig. 2.4. In this representation, the x-axis corresponds to the sentence index, indicating its position within the narrative, while the y-axis represents the size of the vocabulary corpus.

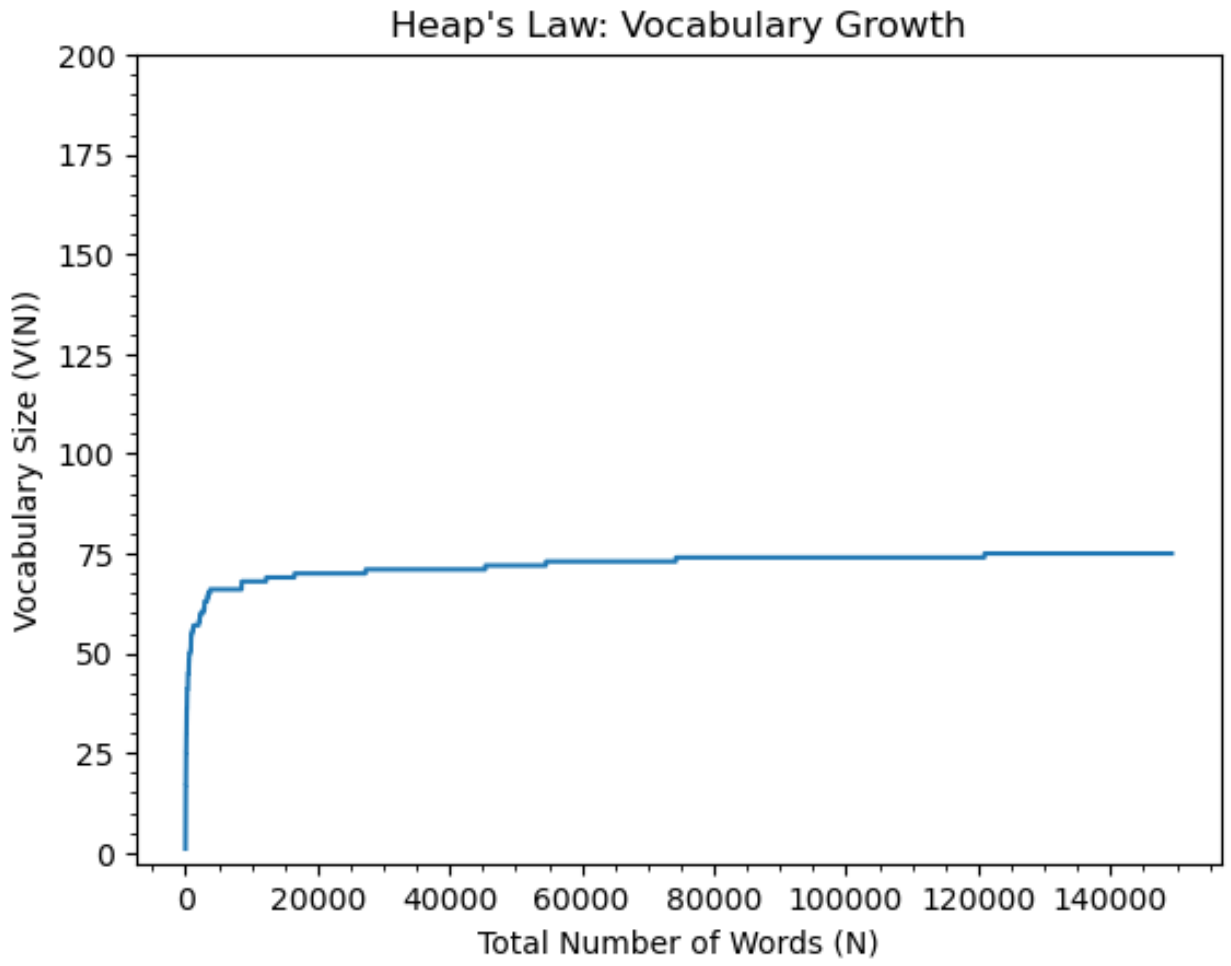


Figure 2.4: Heaps Law graph for *Indiana Jones and the Raiders of the Lost Ark*

As illustrated in Fig. 2.4, the screenplay's vocabulary size is 73 but reaches 69 words or 95% of the vocabulary size at 12204 words. This indicates that most vocabulary is introduced relatively early in the screenplay, reflecting the constrained and purposeful language typical of screenplay writing.

Part Of Speech

In an English sentence, words are classified into various parts of speech, such as nouns, verbs, adjectives, adverbs, conjunctions, and more[5]. Each of these categories plays a specific role in sentence structure and meaning. By leveraging natural language processing libraries like SpaCy[5], I can perform part of speech tagging to identify and label each word according to its grammatical function. Using SpaCy, I can extract two types of part-of-speech tags for each word: a generalized universal part-of-speech tag[6] and a more detailed fine-grained tag[7]. By aggregating this information across an entire screenplay, I can calculate the distribution of each tag type as a percentage of the total word count.

This analysis enables us to gain insights into the compositional makeup of a screenplay through the lens of part of speech. I can characterize the patterns and stylistic choices in the screenplay by analyzing the distribution and frequency of different parts of speech, such as nouns, verbs, adjectives, and adverbs. This in-depth POS analysis also allows us to determine whether the sentence composition of part of speech for generated screenplays aligns with that of human-authored screenplays. Such comparisons can help in assessing the generative ability of the the models, identifying discrepancies or similarities that may indicate the effectiveness of the screenplay generation model in mimicking human-like linguistic behavior.

[5]https://spacy.io/usage/linguistic-features
[6]http://universaldependencies.org/u/pos/
[7]https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Table 2.1: Distribution of Parts of Speech In *Indiana Jones Raiders of the Lost Ark*

Part of Speech	**Values**	**Percent**
adjective	1810.0	0.0681
adposition	3464.0	0.1303
adverb	1452.0	0.0546
auxiliary	1315.0	0.0495
coordinating conjunction	869.0	0.0327
determiner	3446.0	0.1296
interjection	64.0	0.0024
noun	5229.0	0.1967
pronoun	2761.0	0.1039
proper noun	1916.0	0.0721
subordinating conjunction	347.0	0.0131
verb	3910.0	0.1471
Total	26583.0	

As shown in Table 2.1, nouns, pronouns, verbs, adpositions, and determiners are frequently used in screenplays as each occupies at least 10% of the words in the screenplay. This observation underscores the consistent linguistic structure typical of screenplay writing. By analyzing this distribution across multiple screenplays, these common POS categories can serve as a baseline for benchmarking and comparing generated screenplays. Identifying the prevalence and significance of these elements allows us to evaluate whether generated scripts adhere to the linguistic norms of human-authored screenplays or diverge in notable ways.

The same analysis can be done on the universal tags using the more detailed fine-grained tag. The more granular part of speech tag distribution for Indiana Jones and Raiders of the Lost Ark can be found in Table A.1 and Table A.2.

Gini Index

By leveraging part-of-speech tags, I can analyze whether the writer disproportionately indexes a small vocabulary of words in each part-of-speech category of part of speech, offering deeper insights into their linguistic style and choices. One particularly powerful analysis that can be performed using part of speech tags is calculating the Gini Index for each category. The Gini Index is a statistical measure of dispersion that quantifies the degree of inequality in a distribution[11]. By applying the Gini Index, I can assess the imbalance in the usage of specific words within each POS category, revealing patterns of linguistic inequality.

Table 2.2: Gini Index of Parts of Speech for *Indiana Jones Raiders of the Lost Ark*

Part of Speech	**Values**	**Gini**
adjective	1810.0	0.4749
adposition	3464.0	0.6925
adverb	1452.0	0.5338
auxiliary	1315.0	0.7419
coordinating conjunction	869.0	0.4000
determiner	3446.0	0.4819
interjection	64.0	0.0551
noun	5229.0	0.5537
pronoun	2761.0	0.6331
proper noun	1916.0	0.0250
subordinating conjunction	347.0	0.5202
verb	3910.0	0.5066
Total	26583.0	

To conduct this analysis, I calculate the Gini Index[8] for individual POS categories, such as nouns, verbs, adjectives, and adverbs. This involves determining how evenly or unevenly words in this part of the speech category are distributed across the text. A high Gini Index for a particular category indicates that there is a high usage of a few select words in a category, whereas a low Gini Index suggests a more uniform distribution. By systematically examining each part of the speech tag group, I can identify if a writer has an over-reliance on a certain vocabulary. [11]. Table 2.2 reveals that most part-of-speech tags in screenplays exhibit moderate to high Gini index scores. These score found by the Gini index indicates a pronounced inequality in the distribution of words associated with each POS tag, suggesting that certain words are over-indexed within their respective categories. This pattern reflects the repetitive nature of language in screenplays, where specific vocabularies, such as character names, scene descriptors, and common verbs, are used.

Yngves and Frazier Mean Complexity Score

The Yngve and Frazier methods are two complexity scoring techniques developed to measure the syntactic complexity of spoken language. While both approaches rely on syntactic tree representations, they differ significantly in their underlying methodology and focus [5, 12].

Yngve scoring

The Yngve scoring method is based on a stack that tracks the nodes in a syntactic tree as a sentence is processed. This method assigns scores by traversing the Penn Treebark, adding part of speech tag nodes to the stack while moving through the tree from top to bottom and left to right. When a terminal node (a word) is encountered, the size of the stack at that point determines the score for the word. The overall syntactic complexity of the sentence is then calculated as the mean of the individual word scores[12]. This approach focuses on the mechanics of tree traversal and the structural depth at each point without being concerned with the nodes' specific labels or linguistic categories.

[8]https://github.com/oliviaguest/gini/tree/master

Frazier scoring

The Frazier scoring method emphasizes the role of syntactic labels and tree structure. Here, the score for a word is determined by tracing upward through the tree from the terminal node until either the root is reached or the traversal exits the leftmost branch. During this backward traversal, all tags give one point, while sentence and sentence complement nodes give a 1.5 score. This process reflects the syntactic relationships and dependencies encoded in the tree, prioritizing the hierarchical organization and tags of the sentence[12].

Together, these methods offer a nuanced way to analyze syntactic complexity, with Yngve and Frazier scoring useful for finding patterns in sentence structure, the complexity of sentences throughout the screenplay, and variations in sentence construction. To visually represent this analysis, Yngve and Frazier mean scores were plotted as a scatter plot, with each data point corresponding to a sentence in the story as seen in Fig. .5. The x-axis represents the sentence index, indicating its position in the narrative, while the y-axis displays the complexity score. Alternating vertical bands are included to signify transitions between scenes, enhancing the clarity of the scene boundaries.

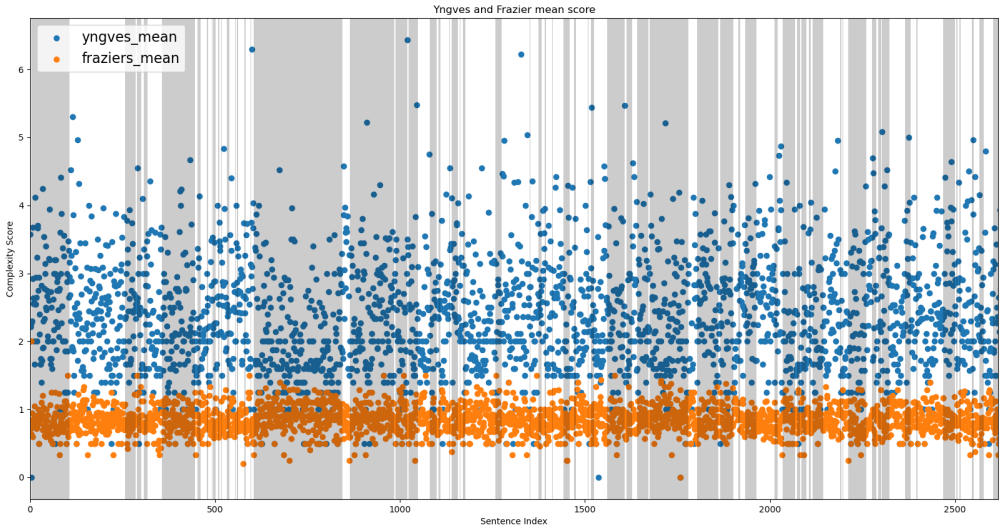


Figure 2.5: Yngves and Frazier Mean for Indiana Jones and the Raiders of the Lost Ark

Through this evaluation, I observe that for Indiana Jones and Raiders of the Lost Ark, the Yngve score has a mean value of approximately 2.33, while the Frazier score averages around 0.85. Additionally, As illustrated in Fig. 2.5 the graph reveals no discernible patterns or trends in sentence complexity for either scoring method, suggesting that sentence in the screenplay is random and does not include repeating sentence pattern.

Scene Length by Sentence

Scene length, measured by the number of sentences, is a visual evaluation method for analyzing the number of sentences in each scene in a screenplay. Analyzing scene length can highlight patterns in the writer's pacing. It may uncover tendencies toward rapid, fragmented storytelling or a consistent pacing with scenes of similar length. For generated screenplays I can assess whether they exhibit variability in scene lengths or if they consistently generate a fixed number of tokens, even when not prompted to. This can provide insight into the diversity and structural complexity of the screenplay, highlighting whether the generation process effectively mimics natural variations in scene length found in human-written scripts.

Two different graphs can be made to visually represent Scene Length by Sentence. The first, as shown in Fig. 2.6, is a scatter plot illustrating scene length across the screenplay. In this representation, the x-axis represents the sentence index, indicating the length of the sentence, while the y-axis shows the frequency of each sentence length occurring in the screenplay. This graph can help identify patterns or trends in sentence length, highlighting whether specific sentence lengths are more prevalent throughout the narrative.

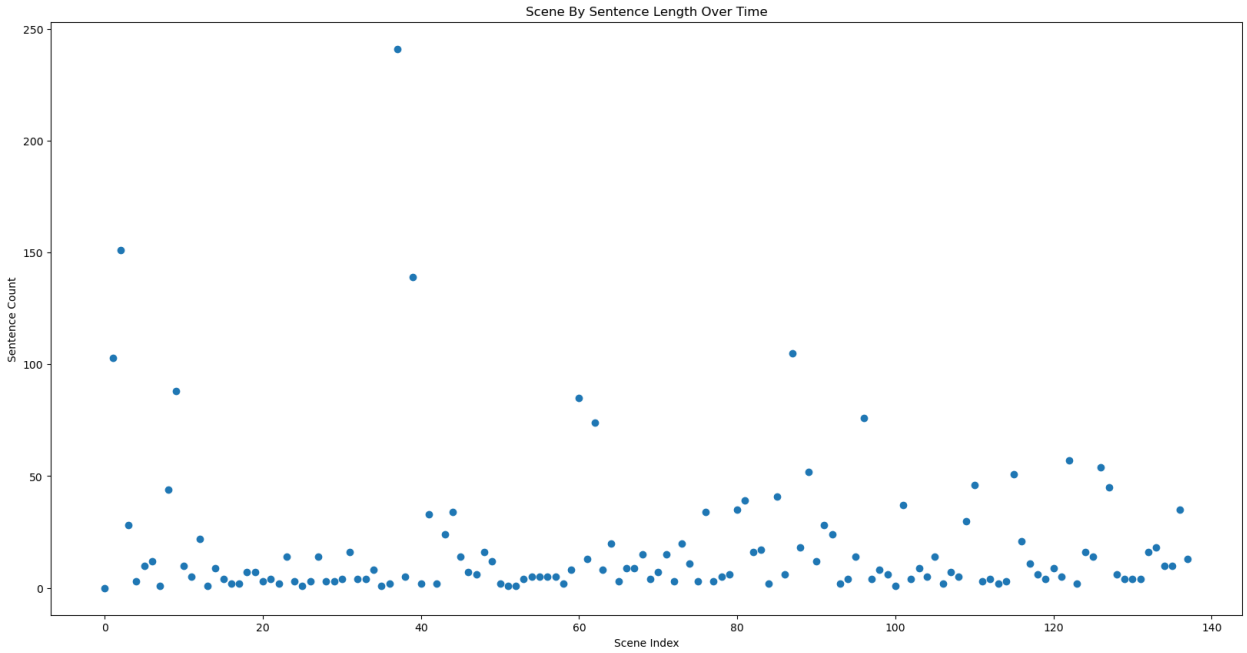


Figure 2.6: Scene By Sentence Length Over Time for *Indiana Jones and the Raiders of the Lost Ark*

The second graph, a histogram shown in Fig. 2.7, represents the aggregate of scene lengths. In this representation, the x-axis represents the sentence index, indicating its position within the narrative, while the y-axis shows the sentence length. This graph can help identify specific points in the screenplay where the writer opted for longer sentences to convey more complex ideas or provide detailed exposition.

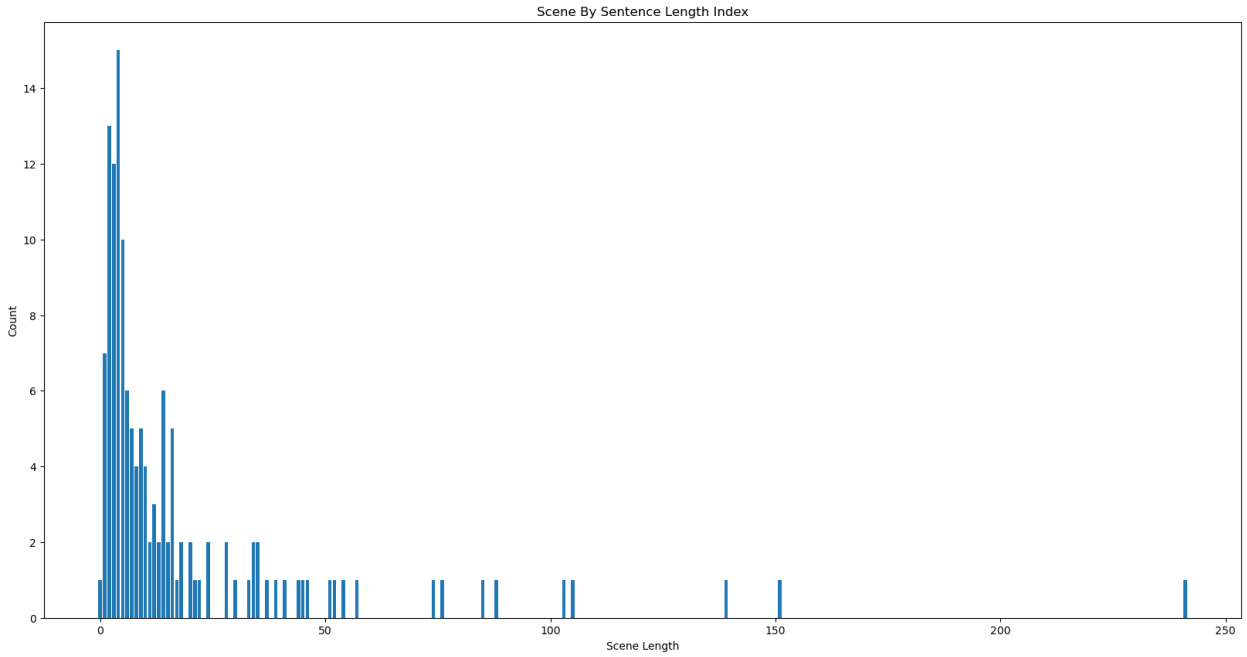


Figure 2.7: Scene By Sentence Length for *Indiana Jones and the Raiders of the Lost Ark*

Sentence Length by Words Count

Sentence length, measured by the number of words, is a complementary visual evaluation method to the Scene Length by Sentence Count analysis. Examining the word count provides a more detailed view of sentence variability, addressing gaps left by sentence count alone. This evaluation method can also help determine whether the patterns observed in the Yngve and Frazier scores are influenced by sentence length, offering additional context for understanding sentence complexity trends within the screenplay.

Like scene length by word count, two different graphs can be made. The first graph, as shown in Fig. 2.8, is a scatter plot illustrating scene length across the screenplay. In this representation, the x-axis represents the sentence index, indicating the length of the sentence by word count, while the y-axis shows the frequency of each sentence length occurring in the screenplay. Alternating vertical bands are included to signify transitions between scenes, enhancing the clarity of the scene boundaries.

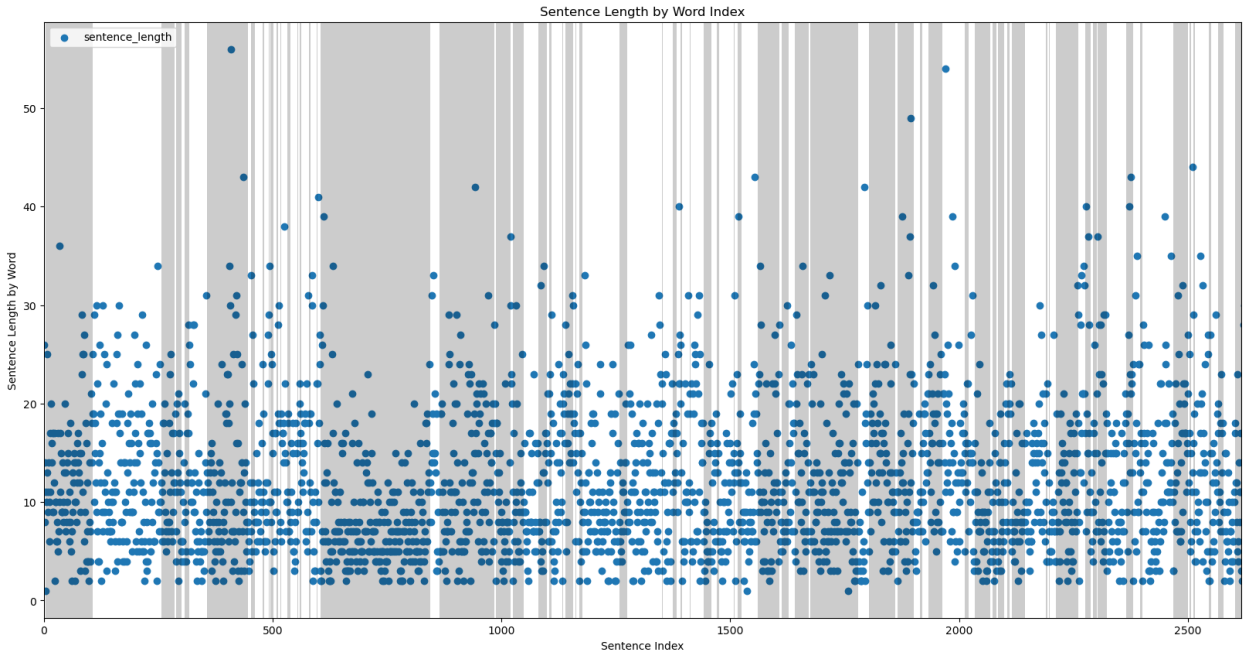


Figure 2.8: Sentence Length by Word Count Over Time for *Indiana Jones and the Raiders of the Lost Ark*

The second graph, as shown in Fig. 2.9, is a scatter plot illustrating scene length across the screenplay. In this representation, the x-axis represents the sentence index, indicating the length of the sentence by word count, while the y-axis shows the frequency of each sentence length occurring in the screenplay.

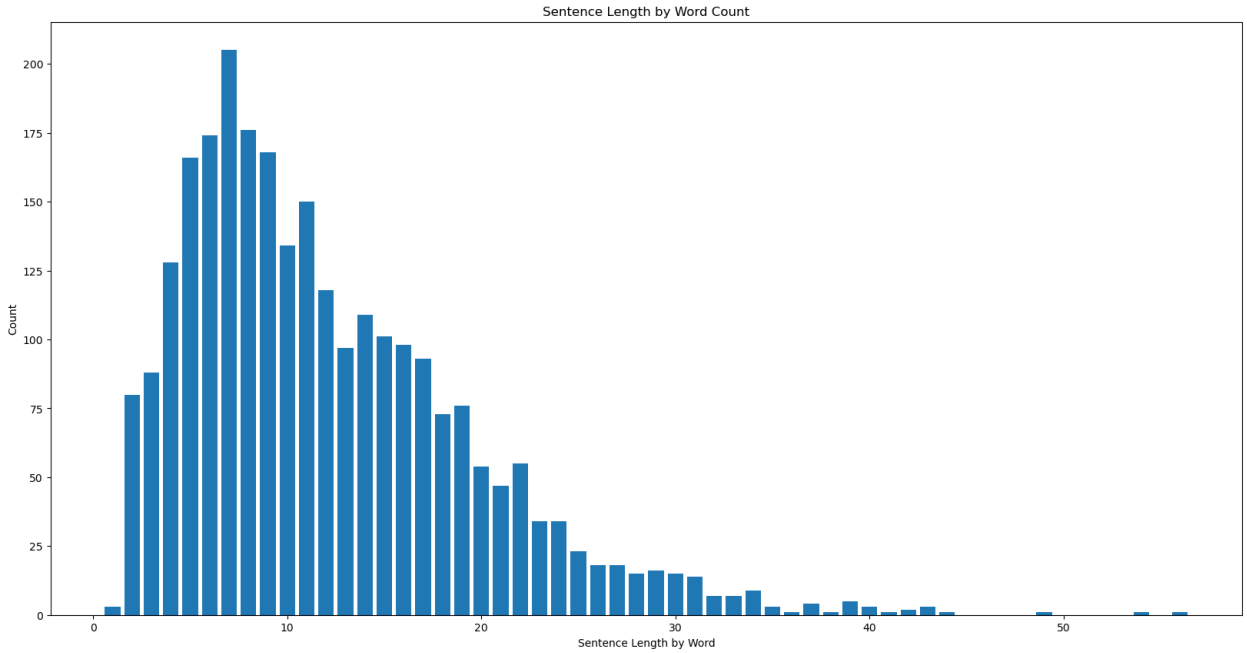


Figure 2.9: Sentence Length by Word Count Indexed for *Indiana Jones and the Raiders of the Lost Ark*

Sentiment Analysis

Sentiment Analysis is a technique from natural language processing used to determine the emotional tone of the underlying text, classifying it as either positive or negative.[5] This process involves analyzing the text to assign each sentence a sentiment score, which reflects the overall tone found in each sentence and section of the screenplay. For this project, I utilized the baseline sentiment analysis pre-trained model from Hugging Face called distilbert-base-uncased-finetuned-sst-2-english, known for its robust and accurate sentiment detection capabilities. Incorporating sentiment analysis into this screenplay evaluator serves multiple purposes. It provides insights into the story's emotional tone overall but can also help identify bias toward one sentiment over the other in the screenplay.

To visually represent this analysis, sentiment scores have been plotted as a scatter plot as seen in Fig, 2.10, with each data point corresponding to a sentence in the story. The x-axis represents the sentence index, indicating its position in the narrative, while the y-axis displays the sentiment

score. Alternating vertical bands are included to signify transitions between scenes, enhancing the clarity of the scene boundaries.

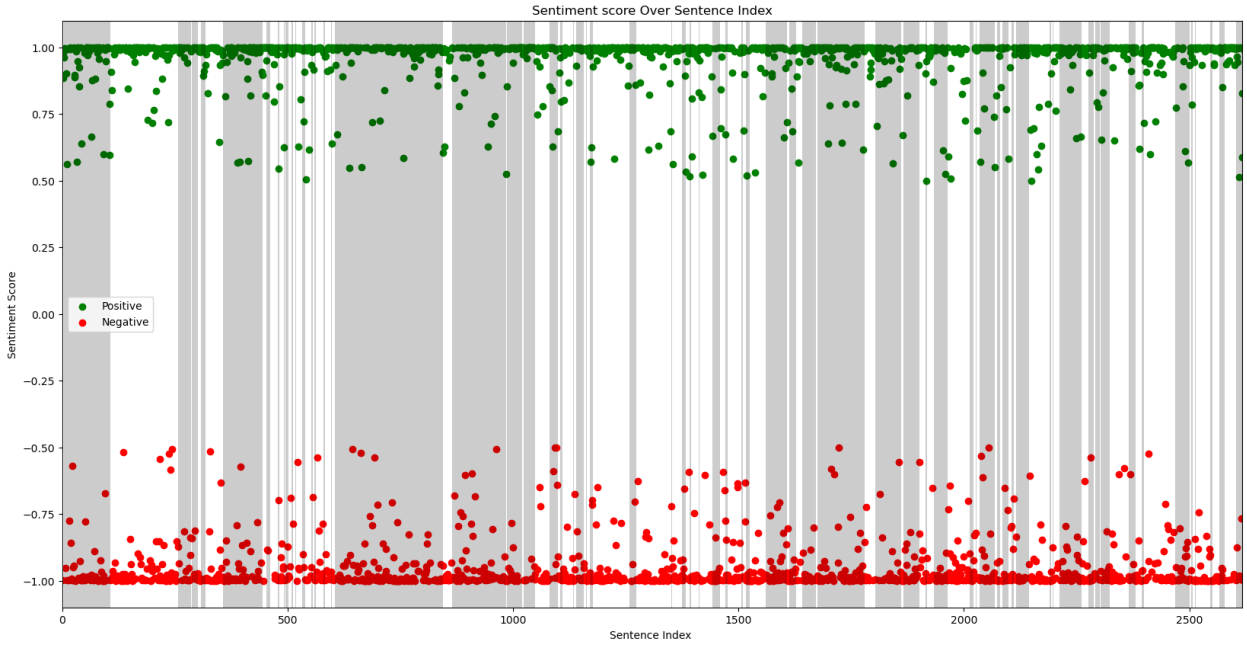


Figure 2.10: Sentiment Analysis for *Indiana Jones and the Raiders of the Lost Ark*

As illustrated in Fig. 2.10 shows that sentiment distribution is relatively balanced, with 1206 sentences displaying positive sentiment and 1306 sentences exhibiting negative sentiment. This even split highlights a neutral tone overall, suggesting a deliberate balance of emotional highs and lows within the screenplay.

Chord Graph

A chord graph is a visualization of the relationship between entities in the screenplay regardless of the scene. In this visualization, characters are arranged around the perimeter of the graph, with chords connecting them to illustrate their interactions. By visualizing relationships across the entire screenplay, the chord graph offers a high-level overview of the frequency of character interactions. Characters with frequent interactions will have more connections, while those with minimal interaction may have fewer connections. In this evaluator, I utilized the HoloViews Bokeh Chord diagram to visualize character interactions, as seen in Fig. 2.11. By processing a co-occurrence

dataframe, I graphed the relationships between characters and represented these connections as chords in the diagram. This approach enabled the creation of an interactive graph that effectively highlights the network of character interactions within the screenplay.

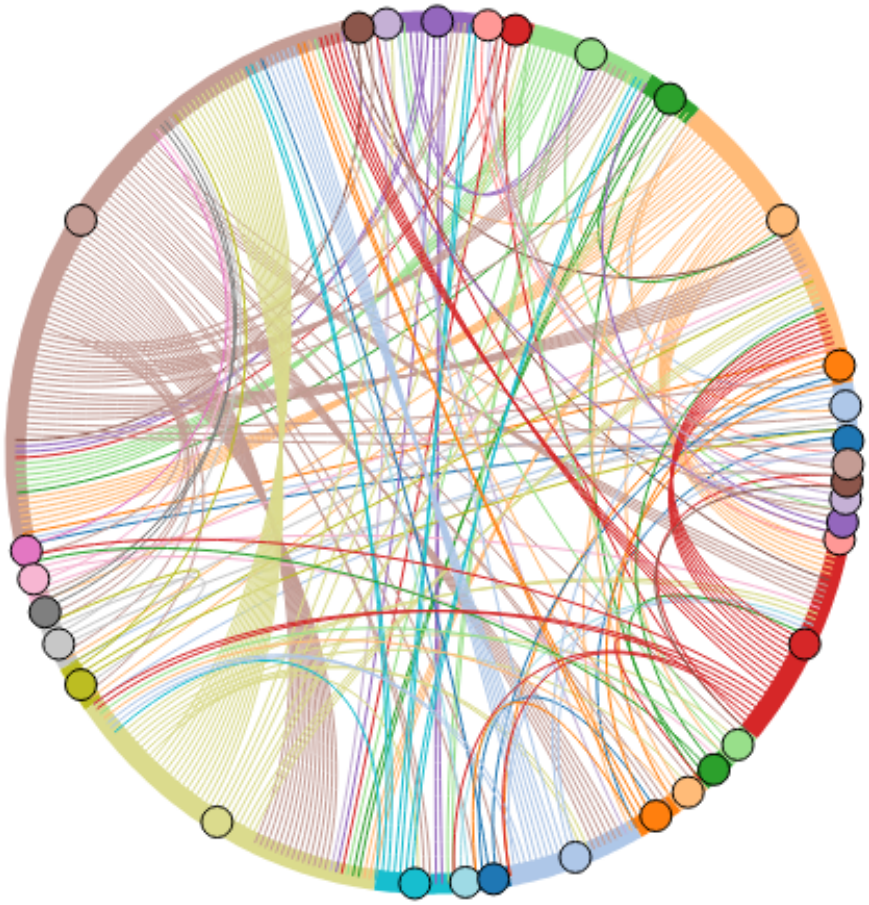


Figure 2.11: Character Co-occurance graph for *Indiana Jones and the Raiders of the Lost Ark*

Forced Directed Graph

A forced-directed graph is a representation of the character's movement through the screenplay. This type of visualization provides valuable insights into how the writer organizes characters spatially and temporally within the story. By observing clusters and connections, one can identify patterns, such as which characters frequently interact, the number of characters present in a scene, and how locations are utilized as narrative devices. Additionally, the graph can reveal thematic structures, such as how movement between locations drives the plot or how certain settings anchor specific groups of characters. Such insights can be instrumental in understanding the underlying narrative framework and the storytelling techniques employed by the writer.

To visualize the forced directed graph I used a simple plot graph, as seen in Fig. 2.12. The x-axis represents the sentence index, indicating its position in the narrative, while the y-axis corresponds to the location. To prevent character nodes in the graph from overlapping, I implemented an offsetting strategy for node positioning. To reduce clutter, scenes with the same heading but different subheadings were considered in the same location to reduce the number of scenes plotted. Additionally, I utilized the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) from SciPy to ensure smooth connections between nodes that do not overshoot the location.

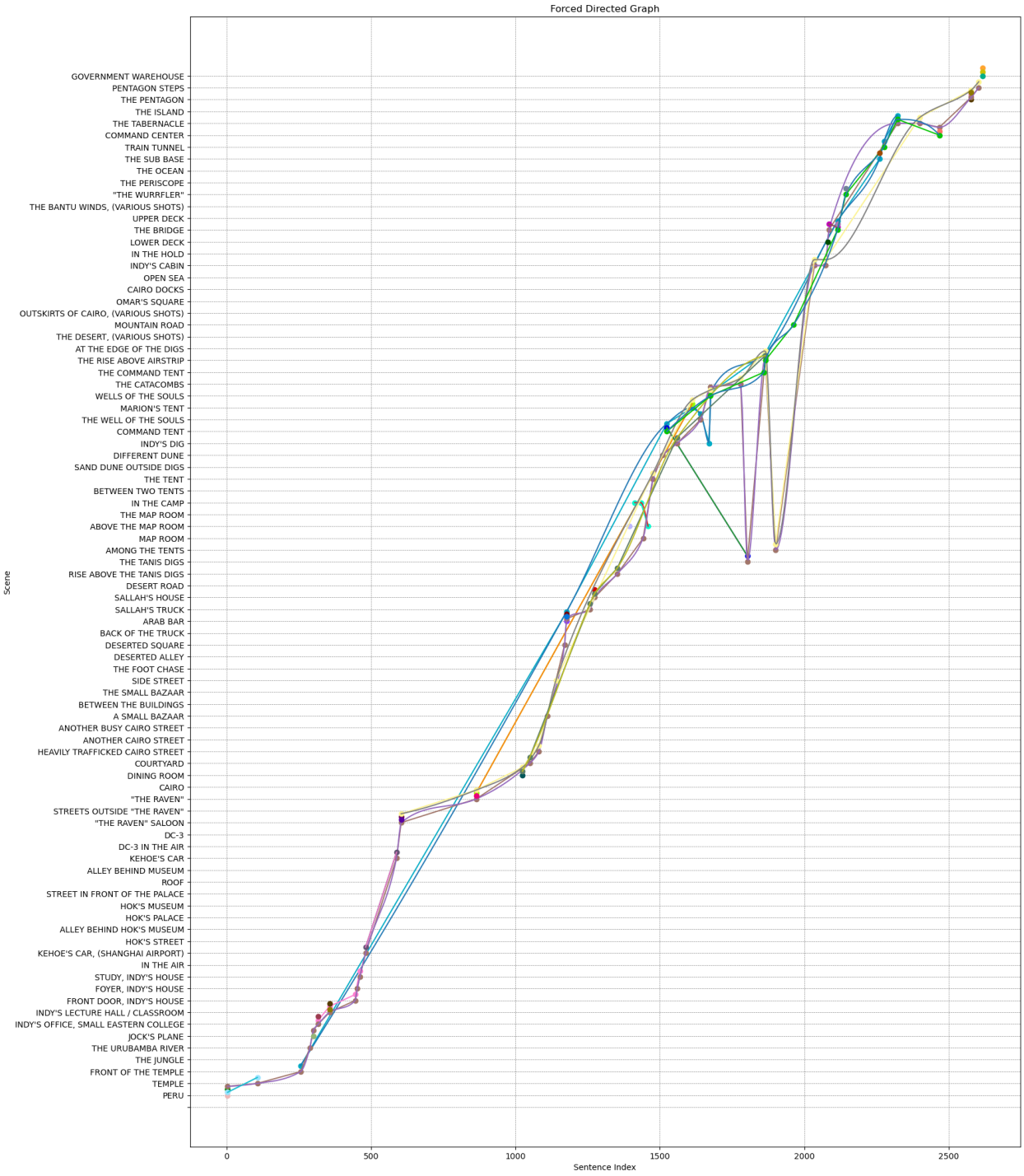


Figure 2.12: Forced Directed Graph for *Indiana Jones and the Raiders of the Lost Ark*

As stated by Akshay Mendhakar and Darshan H S the adventure genre is commonly used to

retell a sequential story and buildup of excitement and relatively little to no flashbacks [14]. This narrative pattern is evident in the directed graph for Indiana Jones and the Raiders of the Lost Ark, which predominantly progresses forward without revisiting previous locations. This forward-moving structure aligns with the genre's emphasis on momentum and the unfolding of events in a linear and engaging manner.

Subject Verb Object Triplets

Subject-verb-object triplets are a method for extracting critical information by identifying the core elements of sentences: the subject (who or what performs an action), the verb (the action itself), and the object (who or what is affected by the action)[6]. This extraction technique highlights the key actors, actions, and objects in a given text, enabling us to analyze the relationships in a. In this thesis, I use D. Rusu, L. Dali, B. Fortuna, and D. Mladenić SVO program [13] [9] to help extract SVO pairs found in the screenplay. However, the extraction is very limited as these were built for regular text, with the subject and object usually being more directly defined rather than implicit like those in screenplays.

[9]https://github.com/HassanElmadany/Extract-SVO

Chapter 3:

Results

In this section, I present the results of our analysis, focusing on the comparative performance between AI-generated screenplays and human-authored screenplays. I analyze multiple generated screenplays and compare their results to human-authored ones. Given the extensive visualizations generated, I will focus on a representative subset to illustrate the key findings. All visualizations generated from the screenplays evaluated in this thesis can be found in the GitHub repository: https://github.com/KnightinGale9/ScreenplayEvaluator/tree/main/ScreenplayUsedinThesis.

The results section compares 12 distinct screenplay generative approaches and a human-authored screenplay section acting as the benchmark. The 12 approaches include each generation method, model, and temperature setting: single-turn and multi-turn generations at temperatures of 0.2, 0.6, and 1.0 using GPT-3.5 Turbo and GPT-4 Turbo. To reduce the influence of individual screenplay variability on quantitative findings, I averaged results across ten screenplays generated by GPT-3.5 Turbo, five screenplays generated by GPT-4 Turbo, and ten human-authored adventure screenplays. The ten human-authored adventure screenplays used in this study include Air Force One, Cast Away, Dumb and Dumber, How to Train Your Dragon, Indiana Jones and the Raiders of the Lost Ark, Jaws, Mission Impossible, Pirates of the Caribbean, Spider-Man, and Star Wars: A New Hope. These screenplays were selected for their diversity in narrative style and genre representation within the adventure category, offering a robust dataset for comparative analysis. By incorporating screenplays written by different authors and representing a variety of subgenres, tones, and storytelling approaches, this selection provides a well-rounded foundation for evaluating the methods and quality of screenplay generation within human-authored works. Therefore, this comprehensive comparison allows us to assess the impact of different generation approaches on the evaluation methods described above for the resulting screenplays.

Generation Testing

Before delving into the evaluation process, it is important to outline the testing conducted to select the appropriate temperature settings and models for screenplay generation. These preliminary tests were crucial for establishing a baseline and ensuring that the chosen configurations yield meaningful and diverse outputs.

The temperature parameter, which controls the randomness of the model's responses, was initially tested at four levels to understand its impact on the generation process. These included a low temperature of 0.2, a medium temperature of 0.6, the default and high temperature of 1, and an additional test at 1.4 to explore the effects of exceeding the default setting. However, when the temperature exceeded 1 and reached 1.4, the generated outputs diverged significantly from the system prompt. Instead of producing coherent screenplay text, the model generated nonsensical content resembling code, as illustrated by the example of a scene's first line provided in A.3. Therefore, the temperature of 1.4 was removed.

During prompt engineering, I found that the specific wording of the prompt was crucial. In early testing, the generated screenplays often exhibited minimal variability in the number of characters and the character names used in popular stories, as illustrated in the Romeo and Juliet example. These screenplays typically featured usually no more than eight characters at a time. To address this, I attempted to include the word "creativity" in the prompt, aiming to generate more diverse and unique outputs. However, this led the model to interpret "creativity" as a directive to enumerate characters, resulting in the presence graph depicted in Fig A.3. This graph shows a sequence of enumerated characters introduced one after another. Given the large number of characters generated, a character legend is provided below to improve readability and facilitate understanding of the character names in Fig A.5. Due to this nature of generation, I removed the word creativity and started shortening the prompt to remove excess words such as natural dialogue.

Finally, I wanted to test both a GPT-3.5 and GPT-4 model to evaluate the evolution in performance in generating screenplays. For the GPT-3 series, GPT-3.5 Turbo was chosen primarily

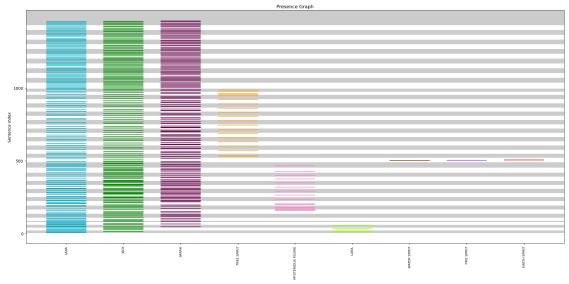
because of its cost-effectiveness in generating tokens and experience using this model. When selecting between different GPT-4 models, there were two primary options: GPT-4-turbo and GPT-4o. Initially, I opted to test GPT-4-turbo, as it was the model I was more familiar with. During testing, GPT-4-turbo consistently generated text that showed results comparable in quality to human-authored content. However, the higher cost of GPT-4-turbo limited the number of generations due to budget constraints. This led to an exploration of whether GPT-4o could deliver similar results at a lower cost, potentially mitigating these cost limitations.

During testing, GPT-4o consistently failed to follow the system prompt as reliably as GPT-4-turbo, requiring major adjustments to the parsing process. For instance, GPT-4o often omitted the standard INT./EXT. format for scene headings, causing the parser to misinterpret these as character names. Additionally, it introduced inconsistencies in transitions and added extra words to character names, necessitating either extensive retooling of the parser or manual post-processing.

Additionally, GPT-4o consistently generated excessively long outputs, producing screenplays exceeding 60,000 words compared to the around 20,000 words typical of GPT-3.5 and around 30,000 words from GPT-4. This inflated length complicated quantitative evaluations between models. The most critical issue arose during the Ygnevz and Frazier mean average testing. At a temperature of 0.2, GPT-4o produced highly repetitive outputs, as illustrated in Figure A.3, highlighting a significant lack of variability and creativity. This performance fell far short of the standards achieved by the GPT-4-turbo model. Given these combined limitations, GPT-4-turbo was selected as the preferred model.

Presence Graph and Increasing Graph

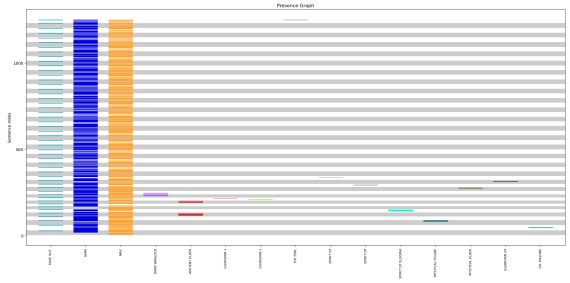
When analyzing character usage in the generative process, I utilize two key visualizations: the presence and increasing graphs. The presence graph illustrates the presence of characters throughout the screenplay, providing a clear view of their involvement and appearance. The increasing graph aggregates the character's appearances over time, showing how their presence evolves as the story progresses.



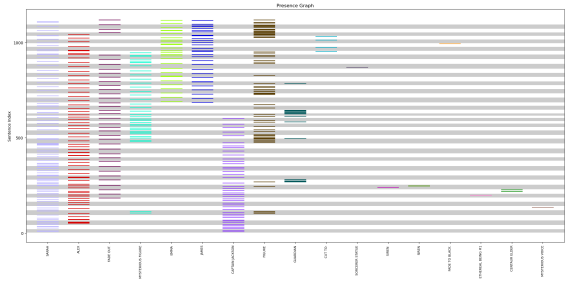
(a) Single Turn GPT 3.5 at 0.2 Temperature

(b) Multi Turn GPT 3.5 at 0.2 Temperature

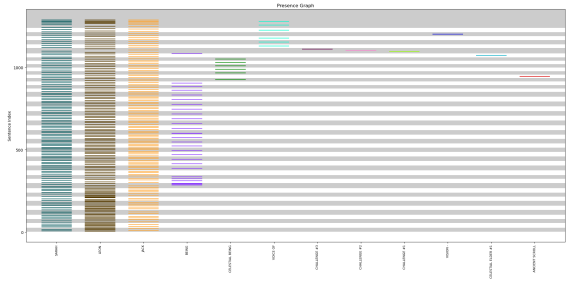
(c) Single Turn GPT 3.5 at 0.6 Temperature



(d) Multi Turn GPT 3.5 at 0.6 Temperature



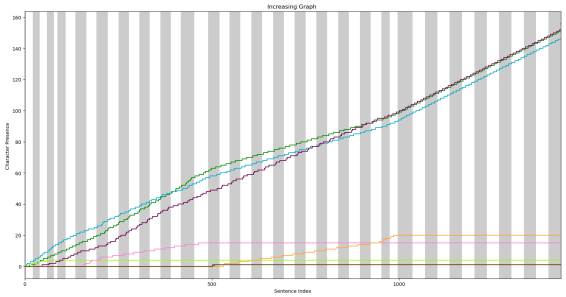
(e) Single Turn GPT 3.5 at 1 Temperature



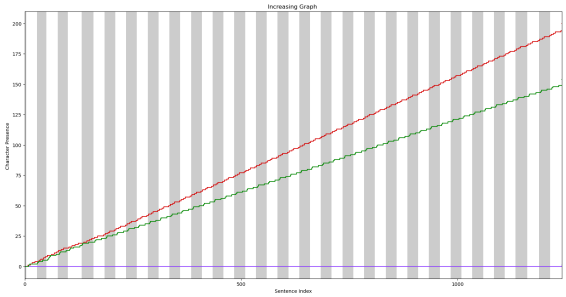
(f) Multi Turn GPT 3.5 at 1 Temperature

Figure 3.1: Presence Graphs from different GPT 3.5 turbo Generation Approaches

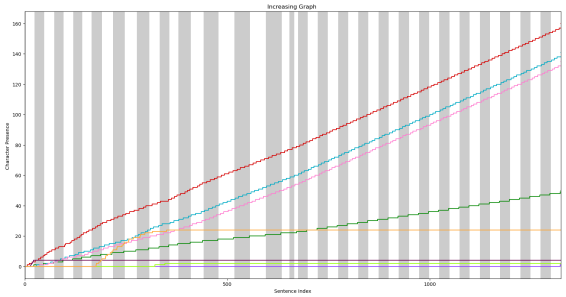
When looking at Fig. 3.1, which contains a selection of the most interesting presence graph, it reveals that GPT writing for the adventure genre of screenplays uses the protagonist-driven narrative structure. As depicted in the figure, the protagonist is consistently present in every scene of the screenplay, underscoring their pivotal role in driving the narrative forward. In contrast, the supporting cast, outside of the other main character, is featured much less frequently, appearing in only a limited number of scenes. This discrepancy suggests that the generated stories emphasize the protagonist's perspective while relegating supporting characters to more peripheral roles.



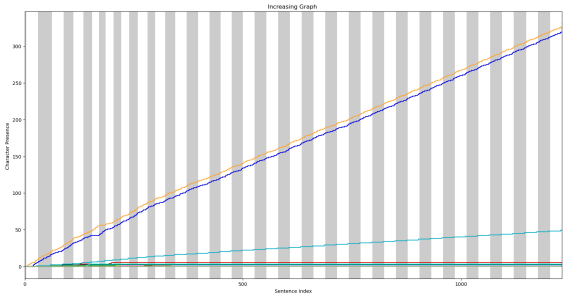
(a) Single Turn GPT 3.5 at 0.2 Temperature



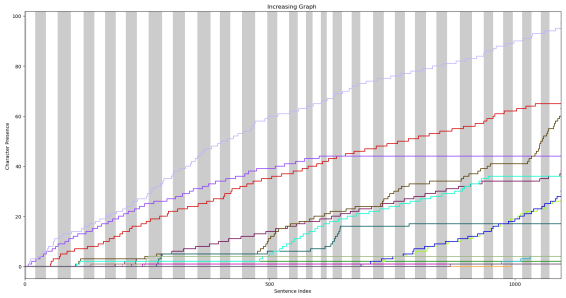
(b) Multi Turn GPT 3.5 at 0.2 Temperature



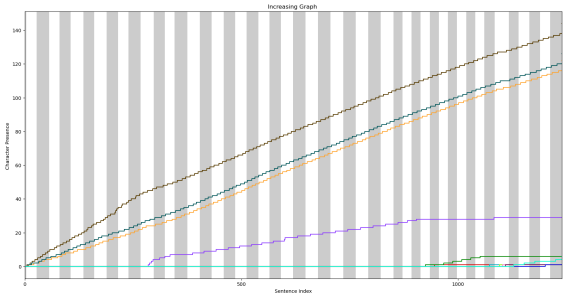
(c) Single Turn GPT 3.5 at 0.6 Temperature



(d) Multi Turn GPT 3.5 at 0.6 Temperature



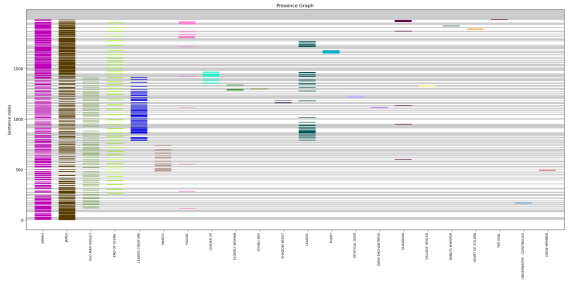
(e) Single Turn GPT 3.5 at 1 Temperature



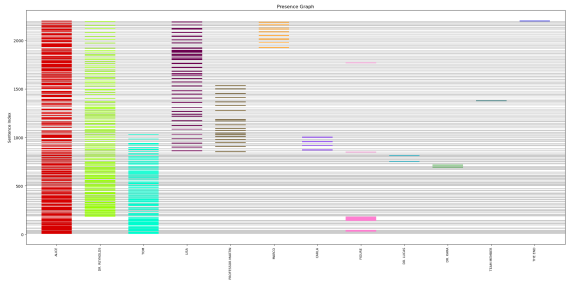
(f) Multi Turn GPT 3.5 at 1 Temperature

Figure 3.2: Increasing Graphs from different GPT 3.5 turbo Generation Approaches

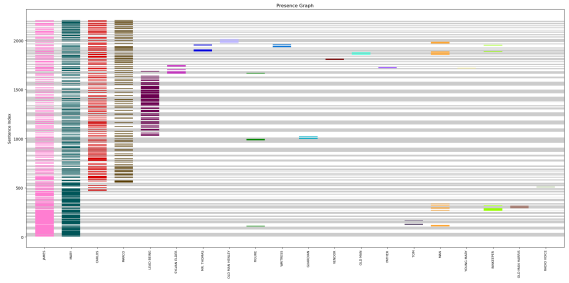
The increasing graph shown in Fig. 3.2 reinforces the evidence from the presence graph, demonstrating that GPT's approach to writing adventure screenplays follows a protagonist-driven narrative structure. In the increasing graph, the top line, representing the protagonist, consistently stands out as the most prominent, highlighting their frequent appearances throughout the screenplay. Meanwhile, the other main characters are represented by lines positioned below the protagonist, and the supporting cast is depicted near the x-axis, reflecting their relatively minimal presence.



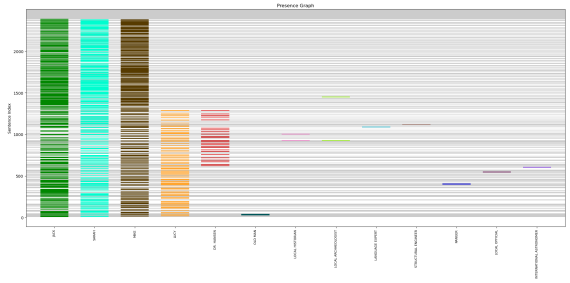
(a) Single Turn GPT 4 at 0.2 Temperature



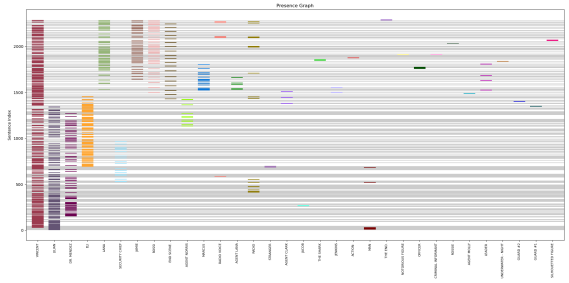
(b) Multi Turn GPT 4 at 0.2 Temperature



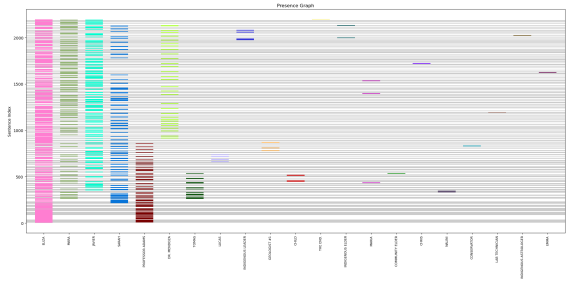
(c) Single Turn GPT 4 at 0.6 Temperature



(d) Multi Turn GPT 4 at 0.6 Temperature



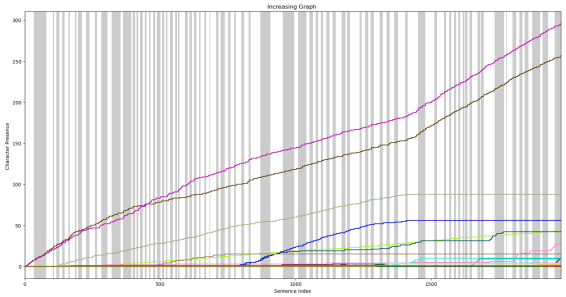
(e) Single Turn GPT 4 at 1 Temperature



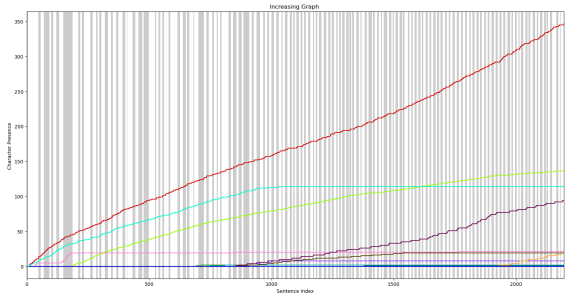
(f) Multi Turn GPT 4 at 1 Temperature

Figure 3.3: Presence Graphs from different GPT 4 turbo Generation Approaches

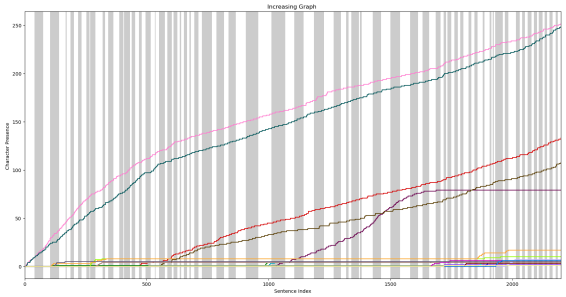
Aside from the information found in the visualization, I find that, on average, GPT-3.5 introduces 9.73 characters per screenplay, with the total ranging from 3 to 22, which is significantly fewer than the average of 50.2 characters found in human-authored screenplays. This contrast highlights the model's limited capacity to create expansive and diverse character ensembles, reflecting a reduced ability to develop intricate character networks often seen in human-authored scripts. This also indicates a significant variability in the model's approach to character generation, though the centrality of a few key characters remains a consistent feature.



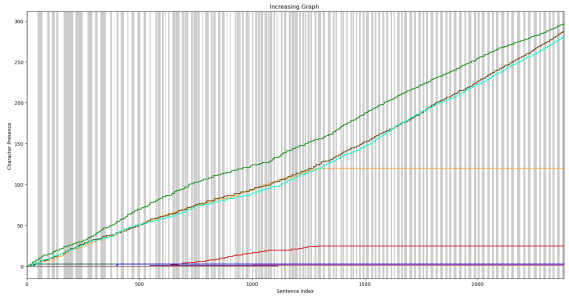
(a) Single Turn GPT 4 at 0.2 Temperature



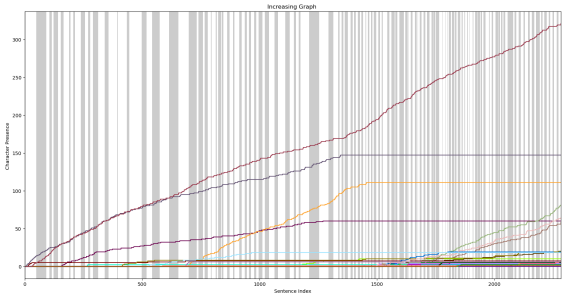
(b) Multi Turn GPT 4 at 0.2 Temperature



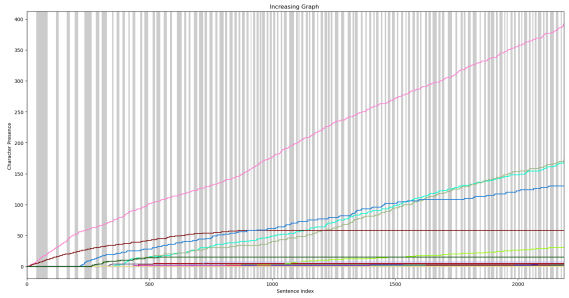
(c) Single Turn GPT 4 at 0.6 Temperature



(d) Multi Turn GPT 4 at 0.6 Temperature



(e) Single Turn GPT 4 at 1 Temperature



(f) Multi Turn GPT 4 at 1 Temperature

Figure 3.4: Presence Graphs from different GPT 4 turbo Generation Approaches

When analyzing the results for GPT-4-generated screenplays, I observe that this model also adheres to a protagonist-driven narrative structure. As shown in Fig. 3.3 and Fig. 3.4, the protagonist remains a central figure, appearing in nearly every scene and consistently represented as the top line in the increasing graph. This highlights the protagonist's prominent role in driving the story forward.

Beyond these visualizations, the presence graph offers an even more compelling insight: GPT-4

demonstrates a markedly expanded and nuanced use of supporting characters. On average, GPT-4 generates 24.7 characters per screenplay, ranging from 11 to 60 characters. While this is fewer than the average number of 50.2 characters in human-authored screenplays, it is approximately 2.66 times more characters than its GPT-3.5 counterpart. This increased size of the supporting cast appearances suggests that GPT-4 constructs a more intricate web of character interactions, enhancing the depth and complexity of its screenplay narratives."

Heaps Law

Table 3.1: Heaps Law Results across all screenplay generation approaches

Model	**Conversation Type**	**Temperature**	**Vocabulary Size**
GPT 3.5 turbo	Single Turn	0.2	53.2
		0.6	57.3
		1.0	59.5
	Multiple Turn	0.2	55.2
		0.6	56.4
		1.0	58.1
GPT 4 turbo	Single Turn	0.2	67.4
		0.6	71.0
		1.0	75.8
	Multiple Turn	0.2	69.2
		0.6	74.4
		1.0	77.2
human-authored			73.8

Heaps' law was applied to evaluate the vocabulary size of the screenplays, providing insights into the richness and diversity of the language employed by different generative approaches. The results, as shown in Table 3.1, reveal distinct trends in vocabulary use. Notably, when comparing screenplays generated by GPT models to their human-authored counterparts, GPT-3.5 exhibited a significantly smaller vocabulary, suggesting less lexical variety. At the same time, GPT-4 demonstrated a vocabulary size comparable to the human-authored screenplays, indicating an improved capacity for diverse language generation.

Across the GPT-3.5 and GPT-4, the type of conversation approach does not appear to impact

the overall vocabulary size significantly. However, there is a correlation between the temperature setting and vocabulary diversity, where increasing the temperature tends to result in a broader and more diverse vocabulary set. Since higher temperature values promote greater variation, they enable the model to explore less common word choices, thereby enhancing lexical diversity. This implies that adjusting the temperature allows one to influence the richness and expressiveness of the output, encouraging the model to take more creative risks in word selection.

Sentiment

Table 3.2: Percentage of Positive Sentiment across all screenplay generation approaches

Model	**Conversation Type**	**Temperature**	**Percentage of Positive Sentiment**
GPT 3.5 turbo	Single Turn	0.2	0.7537
		0.6	0.7718
		1.0	0.7944
	Multiple Turn	0.2	0.9122
		0.6	0.9378
		1.0	0.9394
GPT 4 turbo	Single Turn	0.2	0.7282
		0.6	0.7398
		1.0	0.6360
	Multiple Turn	0.2	0.7436
		0.6	0.7156
		1.0	0.6937
human-authored			0.4808

Sentiment analysis was used as an evaluation metric to test the overall tone of the screenplay. This analysis aimed to determine whether the generated text exhibited any systematic biases in its emotional content, particularly compared to human-authored, human-written text. The results presented in Table 3.2 reveal several trends. When comparing the GPT generative approaches to their human-authored counterpart, I find that the generated text skewed significantly towards positive sentiment. This suggests that the models tend to generate language associated with a positive sentiment across various configurations, indicating an inherent bias towards positivity in their outputs. Such a trend could reflect an optimization bias aiming for user-friendly responses.

Between the different GPT model generation approaches, the most significant contributor to positive sentiment skew was found in the GPT-3.5 model when using a multi-turn conversation style. In this scenario, the proportion of positive sentiment responses consistently remained above 90%, revealing a marked tendency towards overly positive tones. This suggests that GPT-3.5 may overemphasize positive sentiment, especially when engaged in extended interactions, potentially due to how it retains and integrates context from previous turns. The tendency to maintain an overly positive tone may be linked to how the model processes and recalls up to five prior scenes, amplifying positivity in subsequent responses. In contrast, sentiment analysis of the single-turn conversation style for GPT-3.5 revealed a significantly lower positive sentiment proportion, averaging around 77%. This comparative reduction in positive sentiment further proves that an overly positive tone may be linked to a larger context window in multi-turn settings. Without accumulated input from previous interactions, the single-turn setting allows the model to generate responses that are less affected by prior sentiment, which may result in a more balanced output.

A noticeable trend emerges when examining the effects of temperature on the GPT-3.5 model: higher temperature settings generally correlate with a greater proportion of positive sentiment in the generated responses. This suggests that the model tends to favor positive expressions more frequently as the temperature increases, leading to more randomness and creativity in the output. The consistent trend of increased positive sentiment at higher temperatures highlights a potential challenge in maintaining sentiment balance when using GPT-3.5. It suggests that, without carefully calibrating temperature settings, the outputs may lean disproportionately towards positivity, impacting the narrative tone or the intended emotional balance of the generated content.

When examining the GPT-4 generative approaches, the most significant outlier observed was the GPT-4 Turbo model, which, under single-turn conversation settings with a temperature of 1, produced a sentiment score of 63.60 %. This sentiment percentage indicates that, with the more optimized prompts and parameter configurations, it is possible to achieve sentiment distributions that closely resemble those of human-authored, human-written screenplays.

Aside from the outlier when comparing GPT-3.5 and GPT-4 models across different conversation types and temperature settings, I found that GPT-3.5 exhibited a higher sentiment skew in all but one instance. This suggests that GPT-4 consistently delivers a more balanced sentiment distribution, reflecting enhanced nuance and neutrality in its responses compared to GPT-3.5. Notably, this difference is particularly evident in multi-turn conversation styles, where GPT-4 maintains a significantly more balanced sentiment despite the influence of accumulated context from previous interactions. This suggests that GPT-4 is better equipped to handle the complexity of multi-turn dialogues without letting prior conversational turns overly influence the sentiment of its responses.

POS and GINI index

Table 3.3: Comparison of the Percentage of Part of Speech Categories Across GPT-3.5 Generative Approach

Conversation Type	**GPT-3.5 Single Turn**			**GPT-3.5 Multi Turn**			**Human**
Temperature	**0.2**	**0.6**	**1.0**	**0.2**	**0.6**	**1.0**	**Authored**
Part of Speech Distribution							
adjective	0.0622	0.0644	0.0655	0.0554	0.0556	0.0693	0.0547
adposition	0.1281	0.1271	0.1316	0.1294	0.1432	0.1414	0.1142
adverb	0.0417	0.0361	0.0370	0.0197	0.0175	0.0157	0.0496
auxiliary	0.0450	0.0484	0.0443	0.0400	0.0320	0.0285	0.0541
coordinating conjunction	0.0238	0.0263	0.0285	0.0442	0.0485	0.0479	0.0290
determiner	0.1402	0.1454	0.1420	0.1503	0.1511	0.1439	0.1086
interjection	0.0001	0.0002	0.0004	0.0001	0.0001	0.0001	0.0071
noun	0.2264	0.2403	0.2387	0.2459	0.2587	0.2514	0.1973
pronoun	0.1264	0.1194	0.1165	0.1144	0.1071	0.1140	0.1238
proper noun	0.0175	0.0114	0.0192	0.0314	0.0337	0.0331	0.0957
subordinating conjunction	0.0225	0.0197	0.0194	0.0191	0.0196	0.0161	0.0130
verb	0.1661	0.1612	0.1567	0.1502	0.1327	0.1386	0.1529

The use of Part-of-Speech tagging and Gini coefficients for screenplay evaluation offers a unique lens to analyze the structural and stylistic elements of a screenplay. POS tagging enables the identification of linguistic patterns by categorizing words into grammatical classes can reveal the syntactic complexity and the balance between descriptive and action-oriented language. Meanwhile, the Gini coefficient is a measure of inequality in a population. For instance, a high Gini

coefficient might indicate an uneven distribution of words in a part-of-speech category. Together, POS tagging and Gini analysis can provide a quantitative framework to evaluate screenplays, offering insights into the diversity, equity, and stylistic choices embedded in the text.

Table 3.4: Comparison of Gini Index for different GPT 3.5 generation approaches

Conversation Type	**GPT-3.5 Single Turn**			**GPT-3.5 Multi Turn**			**Human Authored**
Temperature	**0.2**	**0.6**	**1.0**	**0.2**	**0.6**	**1.0**	
Gini Coefficients							
adjective	0.6803	0.6256	0.5584	0.7177	0.6582	0.6882	0.4242
adposition	0.6951	0.7312	0.7388	0.7880	0.7866	0.8033	0.7081
adverb	0.6486	0.6134	0.5895	0.6529	0.6109	0.5821	0.5046
auxiliary	0.5861	0.6212	0.6112	0.6971	0.6758	0.6903	0.6656
coordinating conjunction	0.6049	0.6918	0.7062	0.6776	0.6634	0.7258	0.5418
determiner	0.6743	0.5749	0.5604	0.7261	0.7233	0.7045	0.4884
interjection	0.0317	0.0667	0.0538	0.0912	0.0429	0.0250	0.2004
noun	0.6988	0.6804	0.6205	0.7439	0.7065	0.7008	0.4845
pronoun	0.7246	0.7319	0.7086	0.8048	0.7989	0.8053	0.6790
proper noun	0.1131	0.2679	0.2451	0.1878	0.3231	0.2886	0.0549
subordinating conjunction	0.6522	0.5826	0.5624	0.6691	0.6461	0.6533	0.5103
verb	0.6549	0.5939	0.5245	0.6819	0.6209	0.6175	0.4692

When examining the Part of Speech percent distribution for GPT-3.5 in table 3.3, the percentages are largely comparable to those found in human-authored texts across most categories. In single-turn interactions, the only notable deviations occur in the categories of proper nouns and interjections. In multi-turn interactions, deviations are observed in adverbs, auxiliaries, coordinating conjunctions, interjections, and proper nouns. Despite these differences, the overall similarity in part-of-speech distribution suggests that GPT-3.5 effectively captures the natural balance of part-of-speech elements typically seen in human writing. These deviations, while present, are not substantial enough to indicate significant disparities, underscoring the model's ability to generate outputs that closely align with human-like part-of-speech patterns.

When examining trends within the models, I find that temperature settings have minimal impact on certain metrics, but significant differences emerge between conversational styles. Specifically, the conversational style plays a notable role for part-of-speech tags such as adverbs, coordinating

Table 3.5: Comparison of Part of Speech for different GPT 4 generation approaches

Conversation Type	**GPT-4 Single Turn**			**GPT-4 Multi Turn**			**Human Authored**
Temperature	**0.2**	**0.6**	**1.0**	**0.2**	**0.6**	**1.0**	
Part of Speech Distribution							
adjective	0.0657	0.0804	0.0851	0.0868	0.0862	0.0898	0.0547
adposition	0.0903	0.1144	0.1142	0.1123	0.1110	0.1121	0.1142
adverb	0.0365	0.0455	0.0521	0.0403	0.0425	0.0461	0.0496
auxiliary	0.0387	0.0431	0.0366	0.0443	0.0484	0.0437	0.0541
coordinating conjunction	0.0242	0.0318	0.0290	0.0358	0.0336	0.0330	0.0290
determiner	0.1152	0.1415	0.1335	0.1431	0.1371	0.1294	0.1086
interjection	0.0009	0.0012	0.0008	0.0004	0.0008	0.0008	0.0071
noun	0.1993	0.2453	0.2480	0.2552	0.2473	0.2522	0.1973
pronoun	0.0787	0.1005	0.0980	0.0948	0.0968	0.0921	0.1238
proper noun	0.0179	0.0273	0.0298	0.0223	0.0264	0.0318	0.0957
subordinating conjunction	0.0094	0.0136	0.0151	0.0120	0.0134	0.0152	0.0130
verb	0.1229	0.1555	0.1579	0.1526	0.1565	0.1539	0.1529

conjunctions, and proper nouns. In these categories, the single-turn conversational model consistently generates approximately twice as many adverbs and coordinating conjunctions as the multi-turn model. Conversely, the multi-turn conversational style produces roughly twice as many proper nouns, highlighting a clear distinction in linguistic patterns influenced by interaction type.

When examining the GINI index for GPT-3.5 in Table 3.4, I observe that the scores are generally much higher than those of human-authored screenplays, with a few notable exceptions. In single-turn conversations, interjections and auxiliaries are the only categories with smaller GINI index values compared to human-authored counterparts. In multi-turn conversations, interjections are the sole category with a lower GINI index. Apart from these exceptions, all other GINI index scores are comparable to or significantly larger than human-authored screenplays. This suggests that GPT-3.5 tends to exhibit higher variability or imbalance in the distribution of certain linguistic elements compared to human-authored texts.

When examining trends within the models, I find that temperature settings have minimal impact on certain metrics, but significant differences emerge between conversational styles. Specifically, the conversational style plays a notable role for part-of-speech tags such as adverbs, coordinating

Table 3.6: Comparison of Gini Index for different GPT 4 generation approaches

Conversation Type	**GPT-4 Single Turn**			**GPT-4 Multi Turn**			**Human Authored**
Temperature	**0.2**	**0.6**	**1.0**	**0.2**	**0.6**	**1.0**	
Gini Coefficients							
adjective	0.4326	0.5267	0.5167	0.6134	0.5823	0.5519	0.4242
adposition	0.5565	0.7425	0.7394	0.7649	0.7661	0.7591	0.7081
adverb	0.4439	0.5441	0.5352	0.5929	0.5835	0.5465	0.5046
auxiliary	0.5058	0.6133	0.5863	0.6840	0.6374	0.6090	0.6656
coordinating conjunction	0.3419	0.4635	0.5440	0.6191	0.6008	0.5669	0.5418
determiner	0.4845	0.5588	0.5122	0.6667	0.6671	0.5232	0.4884
interjection	0.0836	0.0541	0.0341	0.0567	0.0000	0.1126	0.2004
noun	0.4826	0.5768	0.5555	0.6441	0.6247	0.5888	0.4845
pronoun	0.5270	0.6747	0.6519	0.7117	0.6969	0.6748	0.6790
proper noun	0.0651	0.1231	0.0708	0.1519	0.2928	0.1260	0.0549
subordinating conjunction	0.3973	0.5477	0.5006	0.6225	0.6005	0.5979	0.5103
verb	0.4164	0.5017	0.4609	0.5678	0.5480	0.4848	0.4692

conjunctions, and proper nouns. In these categories, the single-turn conversational model consistently generates approximately twice as many adverbs and coordinating conjunctions as the multi-turn model. Conversely, the multi-turn conversational style produces roughly twice as many proper nouns, highlighting a clear distinction in linguistic patterns influenced by interaction type.

When examining the Part of Speech (POS) percent distribution for GPT- in table 3.5, I find that the percentages are largely comparable to those found in human-authored texts across all categories, except adjective, interjection, and proper noun. Despite these differences, the overall similarity in POS distribution suggests that GPT-4 effectively captures the natural balance of part-of-speech elements typically seen in human writing. These deviations, while present, are not substantial enough to indicate significant disparities, underscoring the model's ability to generate outputs that closely align with human-like part-of-speech patterns.

When examining the GINI index for GPT-4 in Table 3.6, I observe that the scores are generally consistent with those of human-authored screenplays, with a few notable exceptions from individual generative approaches. In single-turn conversation modes with a temperature of 0.2, GPT-4 often generated a lower GINI index than human-authored screenplays in categories such as adpo-

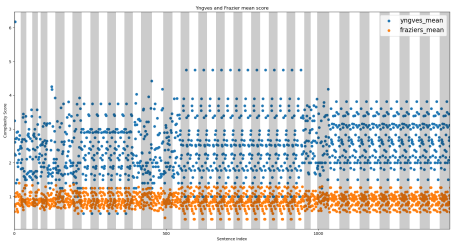
sitions, adverbs, auxiliaries, coordinating conjunctions, interjections, pronouns, and subordinating conjunctions. Beyond these exceptions, the remaining GINI index scores are largely comparable to those of human-authored screenplays. This suggests that GPT-4 generates text with a linguistic balance and equity that closely mirrors the language patterns found in human-written screenplays.

Sentence Complexity

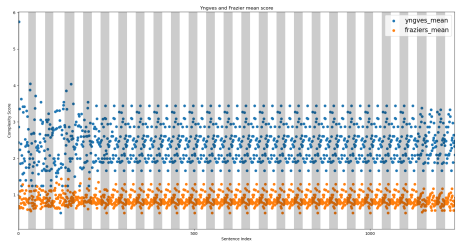
The sentence complexity scores serve as a qualitative measure to assess the intricacy of sentence structures within the text. By analyzing sentence complexity, I can better understand the model's ability to generate linguistically rich text and whether it favors elaborate or straightforward syntax, which can directly impact the quality and engagement of the output. The results presented in Table 3.7 reveal several trends. When comparing the GPT generative approaches to their human-authored counterpart, I find that the generated text produces higher Yngves mean average and Frazier Mean Average scores. This suggests that the AI-generated text generates sentences with more complex syntactic structures compared to human-written screenplays.

Table 3.7: Yngves and Frazier mean average Score for different Generation Approaches

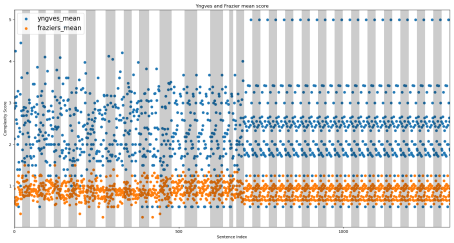
Model	**Conversation Type**	**Temperature**	**Yngves Mean Average**	**Frazier Mean Average**
GPT 3.5 turbo	Single Turn	0.2	2.4011	0.9043
		0.6	2.4117	0.8987
		1.0	2.4931	0.8900
	Multiple Turn	0.2	2.7301	0.8900
		0.6	2.8692	0.8670
		1.0	2.8836	0.8809
GPT 4 turbo	Single Turn	0.2	2.3967	0.8504
		0.6	2.4270	0.8513
		1.0	2.4831	0.8406
	Multiple Turn	0.2	2.5611	0.8392
		0.6	2.5329	0.8466
		1.0	2.5552	0.8345
human-authored			2.0950	0.8540



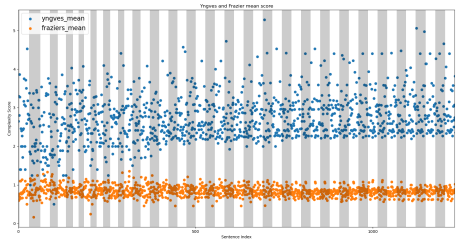
(a) Single Turn GPT 3.5 at 0.2 Temperature



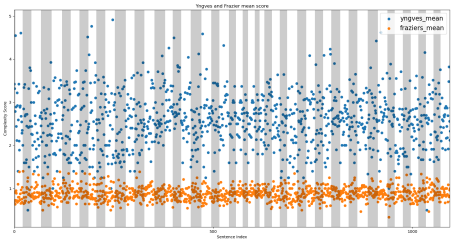
(b) Multi Turn GPT 3.5 at 0.2 Temperature



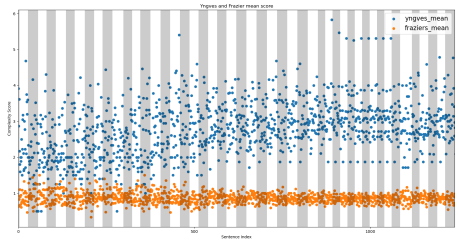
(c) Single Turn GPT 3.5 at 0.6 Temperature



(d) Multi Turn GPT 3.5 at 0.6 Temperature

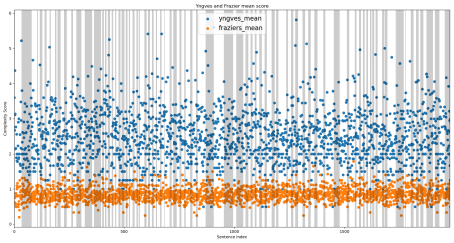


(e) Single Turn GPT 3.5 at 1 Temperature

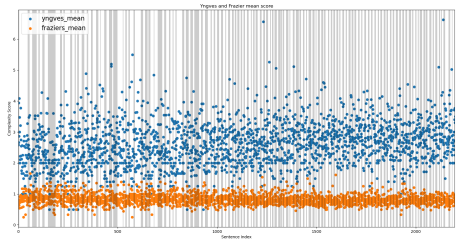


(f) Multi Turn GPT 3.5 at 1 Temperature

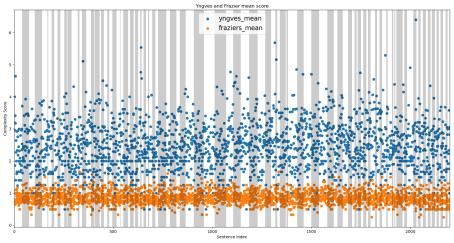
Figure 3.5: Yngves and Frazier mean Average Score Over Time from different GPT 3.5 turbo Generation Approaches



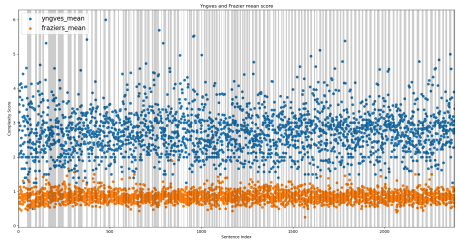
(a) Single Turn GPT 4 at 0.2 Temperature



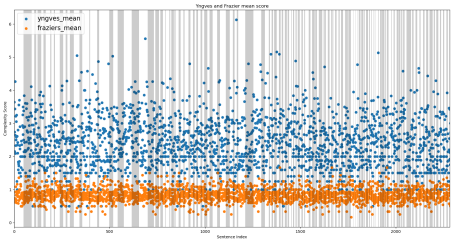
(b) Multi Turn GPT 4 at 0.2 Temperature



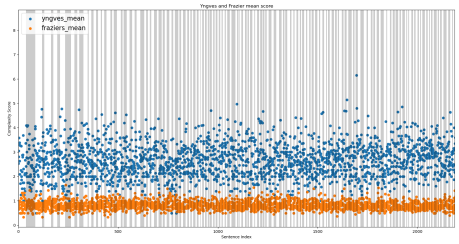
(c) Single Turn GPT 4 at 0.6 Temperature



(d) Multi Turn GPT 4 at 0.6 Temperature



(e) Single Turn GPT 4 at 1 Temperature



(f) Multi Turn GPT 4 at 1 Temperature

Figure 3.6: Yngves and Frazier mean Average Score Over Time from different GPT 4 turbo Generation Approaches

For the average scores of Yngve's mean and Frazier mean, I observe that Yngve's complexity score demonstrates an increase in correlation with temperature while Frazier mean has no strong correlation. This finding suggests that as the temperature parameter rises, which introduces greater randomness and creativity into the model's output, the sentences generated by the model exhibit significantly higher syntactical complexity. This trend indicates that a higher temperature enhances the diversity of word choices and sentence structures and leads to more intricate syntactic relation-

ships within sentences. Another trend in the graph is the consistent difference in complexity scores between GPT-3.5 and GPT-4 models. In nearly all cases, GPT-3.5 demonstrates higher syntactic complexity than its GPT-4 counterparts, with only one exception being the single turn 0.6 approach across the analyzed instances. This difference becomes even more pronounced in multi-turn settings, where GPT-3.5 consistently produces sentences with Yngves scores 0.3 higher than their GPT 4 counterpart. This suggests that GPT-3.5 may favor more elaborate sentence constructions, particularly in extended conversational contexts, whereas GPT-4 appears to adopt a relatively more measured and streamlined approach to sentence generation, possibly favoring clarity and cohesion over raw complexity. These findings underscore key differences in how the two models handle language generation, particularly in scenarios involving sustained dialogue or narrative development. However, having a higher complexity score does not necessarily result in a more cohesive or coherent story, as increased syntactic complexity can sometimes lead to unintended consequences, such as long, convoluted, or tangential sentences. This can be seen in Roberto Franzosi's work as when he applied Yngves score to his work. He found the sentence with the highest complexity score was a long sentence made up of connected clauses that could stand as five independent sentences: "I had taken off below take-off speed, I had survived a crashing bounce back to earth, I had smashed through trees, I had nearly rolled over on our back, and then I had survived a crash landing through fire." .[5]

Aside from the average scores, a more compelling application of these evaluators is the analysis of Yngve and Frazier mean average scores over time. As illustrated in Figure 3.5, distinct repeating patterns emerge in these scores, revealing an important aspect of the GPT model's language generation behavior. This repeating pattern phenomenon is most evident in Fig. 3.5a and Fig. 3.5b, which represent models with the lowest temperature or most deterministic tested setting of 0.2. These patterns suggest that while the generated text may not be textual identical, it often exhibits syntactic similarity, implying a structural repetition within the output of GPT 3.5. When introducing different temperature settings, another important trend emerges the relationship between temperature

and the occurrence of repeating sections in Yngve and Frazier scores. Lower temperature settings generally always produce more deterministic outputs, correlating with increased repetition. For a mid-range temperature of 0.6, repetition is still observed; however, its severity varies. In Fig 3.5c, the level of repetition is comparable to that of the 0.2 temperature setting, while Fig. 3.5d demonstrates almost no repeating sections. This variability highlights that even mid-range temperatures can sometimes exhibit high levels of structural repetition.

Finally, a high temperature of 1 demonstrates the least repetition, producing more varied and less deterministic outputs. Since higher temperatures encourage greater diversity in syntactic structures, it reduces the probability of creating repetitive patterns. Overall, the findings illustrate how temperature settings directly influence the balance between determinism and variability in GPT-3.5's outputs, with higher temperatures fostering more creative and dynamic language generation.

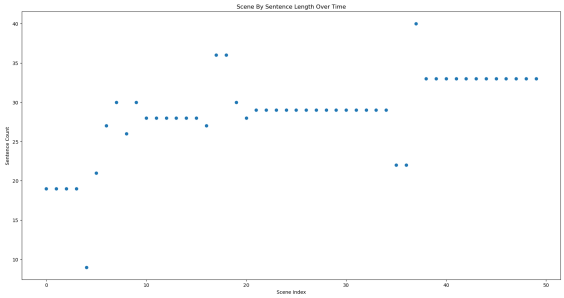
However, when examining the graphs generated by GPT-4 models, it becomes apparent that these repetitions are virtually eliminated, and no discernible patterns of structural repetition can be identified, as seen in Fig 3.6. Visually, the figures' graphs look much closer to those generated by human authors, such as fig 2.5. This stark contrast suggests that GPT-4 models represent a significant leap forward in language generation capabilities compared to their GPT-3.5 counterparts.

The absence of repetitive syntactic patterns in GPT-4 outputs indicates a higher degree of variety in the syntactical build of each sentence in its underlying architecture. This improvement could be attributed to enhanced model optimization and more refined methodologies for controlling temperature and randomness to reduce repetitive text generation. The results suggest that GPT-4 is better equipped to balance consistency and diversity, enabling it to generate syntactically varied and semantically coherent text.

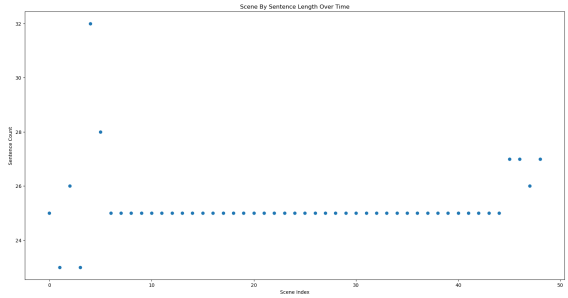
Scene and Sentence Length Analysis

To thoroughly analyze screenplay structure, both scene length and sentence length offer valuable insights into narrative pacing and variability. Scene length, measured by the number of sentences, is a visual evaluation method to assess the density of dialogue and action within each scene.

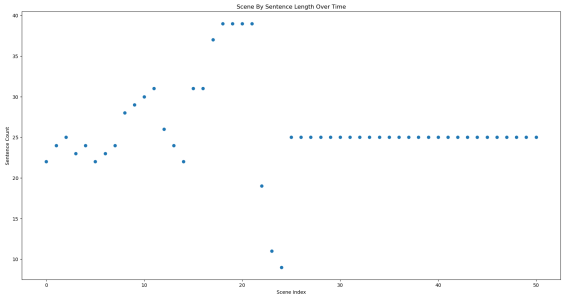
Sentence length, defined by the number of words per sentence, provides a deeper understanding of variability and style, complementing the broader patterns revealed by sentence count alone.



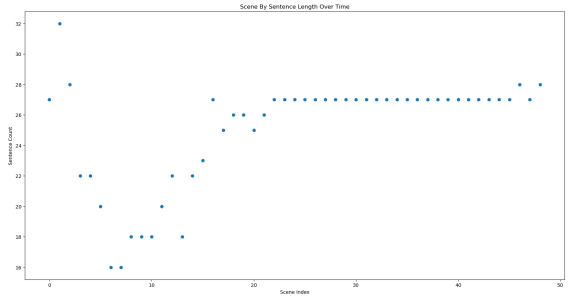
(a) Single Turn GPT 3.5 at 0.2 Temperature



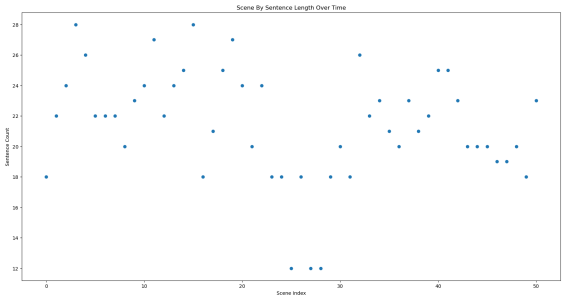
(b) Multi Turn GPT 3.5 at 0.2 Temperature



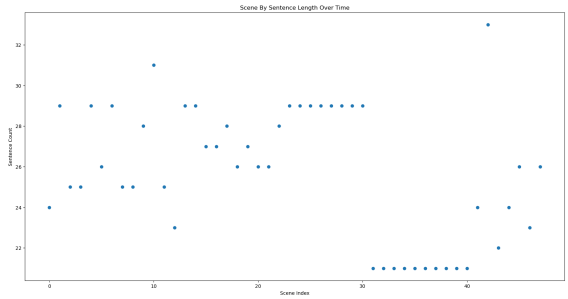
(c) Single Turn GPT 3.5 at 0.6 Temperature



(d) Multi Turn GPT 3.5 at 0.6 Temperature

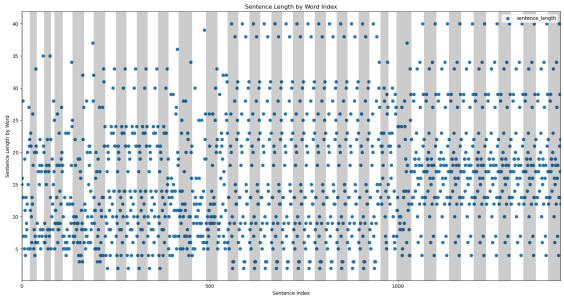


(e) Single Turn GPT 3.5 at 1 Temperature

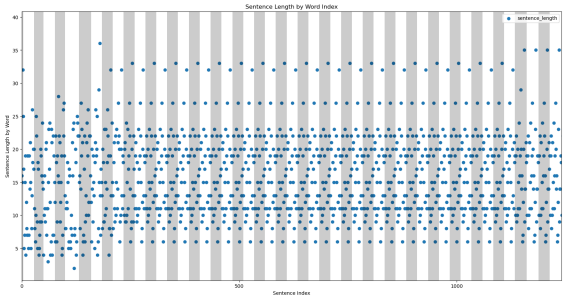


(f) Multi Turn GPT 3.5 at 1 Temperature

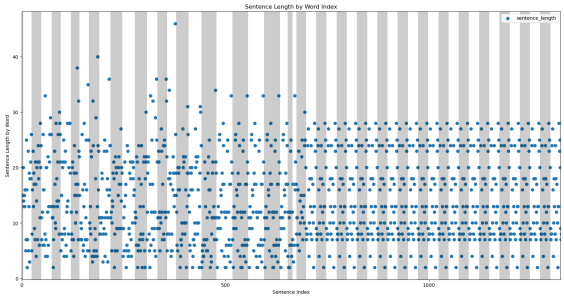
Figure 3.7: Scene Length By Sentence Count Over Time from different GPT 3.5 turbo Generation Approaches



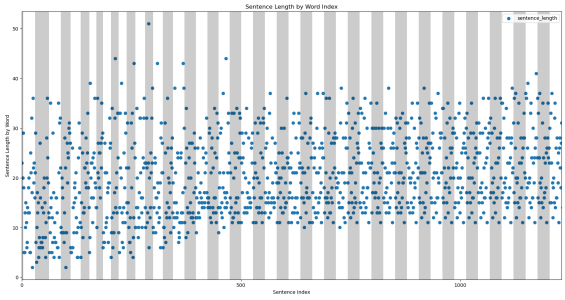
(a) Single Turn GPT 3.5 at 0.2 Temperature



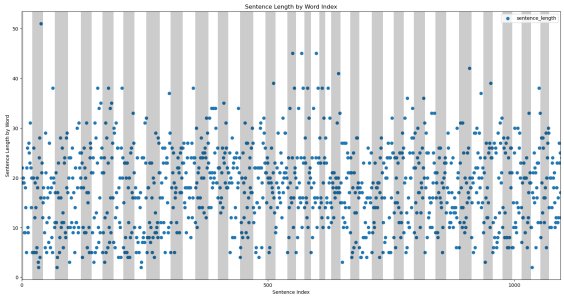
(b) Multi Turn GPT 3.5 at 0.2 Temperature



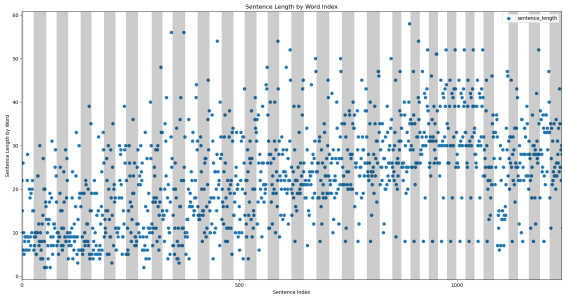
(c) Single Turn GPT 3.5 at 0.6 Temperature



(d) Multi Turn GPT 3.5 at 0.6 Temperature

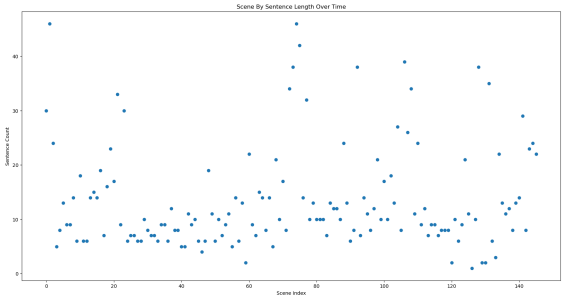


(e) Single Turn GPT 3.5 at 1 Temperature

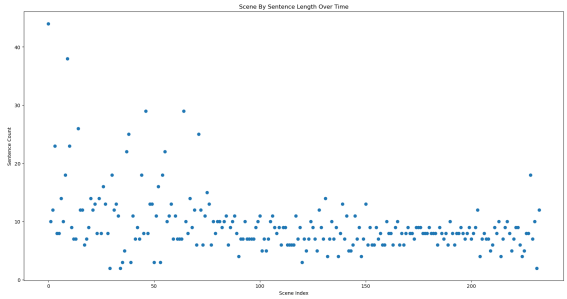


(f) Multi Turn GPT 3.5 at 1 Temperature

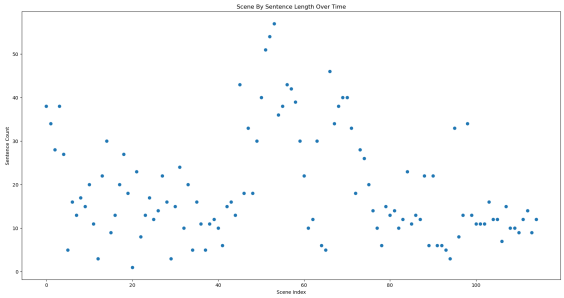
Figure 3.8: Sentence Length By Word Count Over Time from different GPT 3.5 turbo Generation Approaches



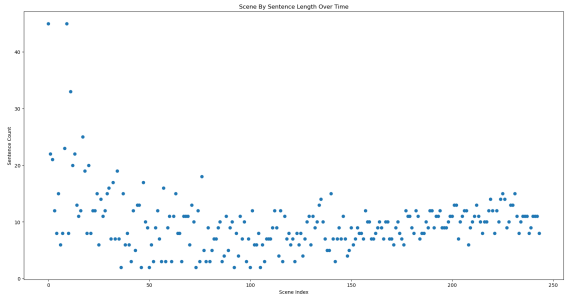
(a) Single Turn GPT 4 at 0.2 Temperature



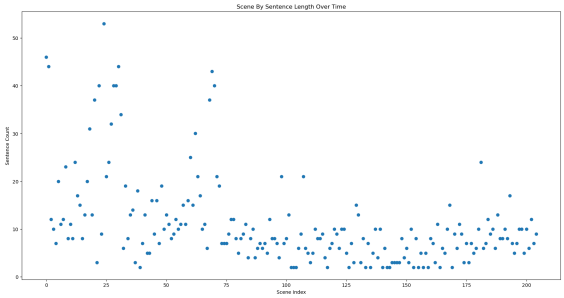
(b) Multi Turn GPT 4 at 0.2 Temperature



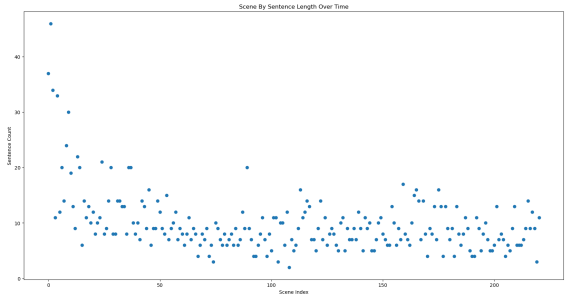
(c) Single Turn GPT 4 at 0.6 Temperature



(d) Multi Turn GPT 4 at 0.6 Temperature

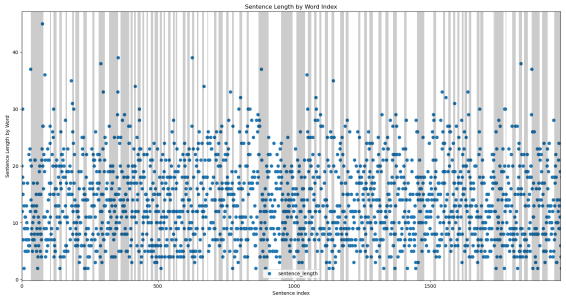


(e) Single Turn GPT 4 at 1 Temperature

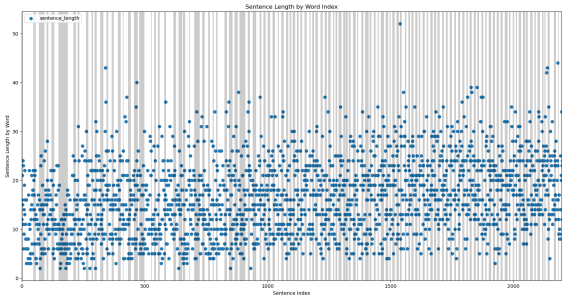


(f) Multi Turn GPT 4 at 1 Temperature

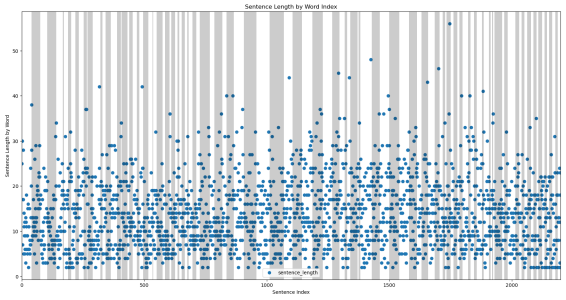
Figure 3.9: Scene Length By Sentence Count Over Time from different GPT 4 turbo Generation Approaches



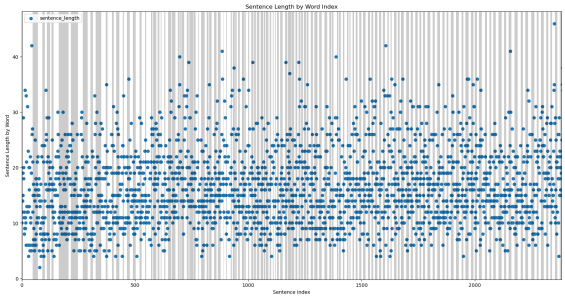
(a) Single Turn GPT 4 at 0.2 Temperature



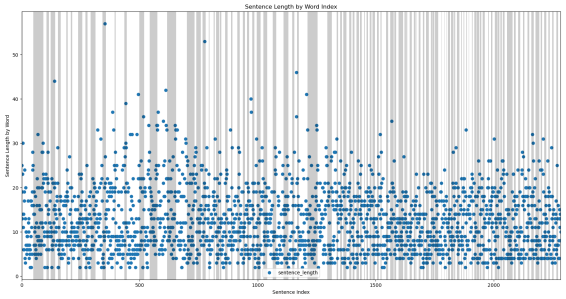
(b) Multi Turn GPT 4 at 0.2 Temperature



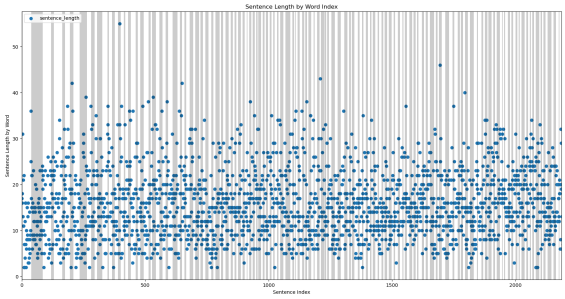
(c) Single Turn GPT 4 at 0.6 Temperature



(d) Multi Turn GPT 4 at 0.6 Temperature



(e) Single Turn GPT 4 at 1 Temperature



(f) Multi Turn GPT 4 at 1 Temperature

Figure 3.10: Sentence Length By Word Count Over Time from different GPT 4 turbo Generation Approaches

In Figure 3.7, I observe that GPT-3.5 generates scenes with noticeable variations in length, typically averaging around 51.1 scenes per generation. Occasionally, GPT-3.5 adds or omits scenes, leading to fluctuations in the total count across different outputs. By examining the scene lengths, I find that while there is some variability in the number of sentences per scene in a screenplay, the generation occasionally falls into a repetitive pattern, limiting the diversity and dynamism of the narrative structure. By looking at where the repetitive patterns occur, I find they seemingly match

up with when Yngves and Frazier score produce repetitive patterns. The worse offender can be seen in Fig. 3.7b where most of the screenplay, this repetition is further illustrated in the number of sentences per scene, suggesting that GPT-3.5 consistently chooses to generate scenes that are similar not only in syntactical complexity but sentence count. This correlation further supports the idea that GPT-3.5's deterministic nature, especially at lower temperatures, can functionally generate the same scene over and over again rather than a new scene.

With this finding, it would be prudent to delve deeper into the scene's structure and analyze the sentence length structure through word count. By shifting the focus to sentence length regarding word count, I can better understand how repetition manifests in the generated content.

When examining the visualization in Fig. 3.8, I observe the recurrence of a repetitive pattern. Using the alternating bands to represent scene boundaries, I find that while sentence length by word count is typically quite varied, it exhibits a repeating pattern that aligns with the repetitive trends identified in Yngve's and Frazier mean average scores and in scene length by sentence count.

This consistency in sentence construction suggests that GPT-3.5 heavily relies on pre-learned patterns or templates during the generation process. Although the model changes the specific words used, it often preserves the underlying sentence structure, leading to a lack of diversity in the overall narrative composition. Such findings highlight a potential limitation in the model's ability to generate more dynamically structured content.

In comparison, GPT-4 exhibits noticeably different behavior when tasked with generating scenes. When given a prompt specifying 50 scenes, GPT-4 often exceeds this number, frequently producing more scenes than initially requested. In this study, I found when asked to generate 50 scenes, it instead generated, on average, 204.7 scenes with a range of 91 to 304 scenes. This tendency to over-generate suggests that GPT-4 may prioritize expanding upon the input to achieve greater narrative depth or to include additional details that enhance continuity and coherence. By doing so, GPT-4 often delivers a richer and more nuanced narrative, albeit at the cost of adhering strictly to the original guidelines.

The visualization from Fig, 3.9 and Fig. 3.10 underscores significant differences between the two models, particularly in handling structural variability within screenplays. Unlike GPT-3.5, GPT-4 exhibits far less deterministic behavior regarding scene sentence counts and sentence lengths by word count. This distinction remains evident even when both models are set to operate at lower temperature settings, which typically favor more conservative and predictable outputs.

This observation suggests that GPT-4's underlying architecture is inherently more flexible and less bound by repetitive or formulaic patterns. Its generative process appears to allow for greater adaptability, enabling the model to deviate more effectively from rigid structures while maintaining coherence.

Forced Directed Graph

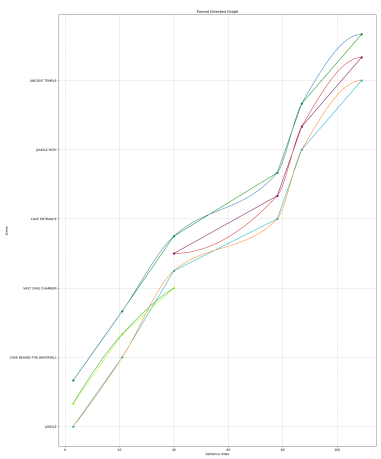
The force-directed graph is a valuable tool for visualizing the movement of characters across various locations within a story. This graph effectively illustrates spatial and narrative relationships by mapping characters and locations as interconnected nodes and edges. As depicted in Figure X, the graphs generated from the selected screenplays exhibit a general trend of increasing connections over time, which aligns with the narrative build-up and complexity typically expected within the genre. This progression reflects the expected pacing and expansion of the storyline as characters interact with more locations or transition to new settings.

This analysis also underscores a significant limitation in GPT-generated screenplays: the lack of diversity and creativity in using locations. The limited diversity is evident in the y-axis of the force-directed graph, representing all unique screenplay locations. A relatively small range on the y-axis indicates a low number of distinct scenes, suggesting that GPT tends to rely on a restricted set of settings rather than creating a rich and varied narrative environment.

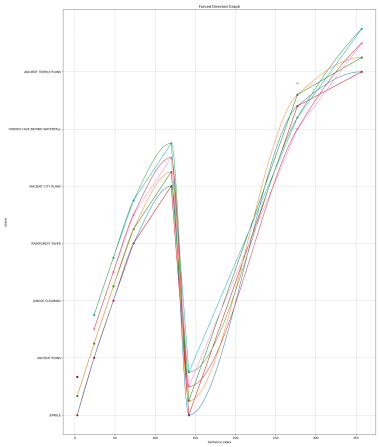
Creativity, or the lack thereof, can also be evaluated through the complexity of the generated Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) lines, representing the trajectories of character movements and transitions between locations. Most of these graphs exhibit functionally linear or simplistic cubic patterns, trending upward and to the right. This indicates a linear and for-

mulaic progression in the narrative's spatial structure where all characters move in unison through locations. However, some graphs occasionally exhibit a hint of complexity, such as repeated returns to previous locations, as demonstrated in Fig 3.11d. This analysis suggests that while GPT-3.5 can model basic spatial movement, it struggles to introduce more dynamic or complex shifts in location transitions, steering them toward simpler screenplay structures.

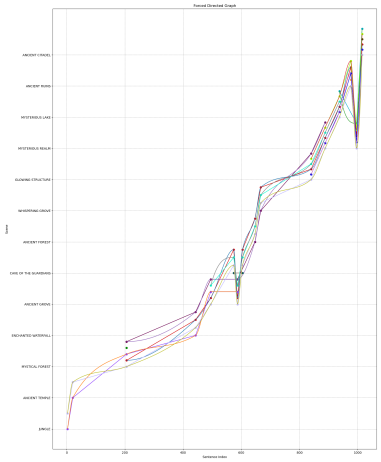
However, with GPT-4, I observe that the force-directed graphs generated are notably more diverse and complex than their GPT-3.5 counterparts. Since I can use the y-axis to track locations, I see that, on average, the location count is higher than those found in GPT 3.5. Though this increase can be partially attributed to GPT-4's tendency to generate more scenes than requested, allowing it to establish a more dynamic and well-paced progression of events, it shows it is less constricted than GPT-3.5 when generating locations. Unlike its predecessor, which often adheres to a rigid pattern in location choice and often has multiple sequential scenes using the exact location, GPT-4 demonstrates greater flexibility and variation, resulting in a more nuanced and less formulaic narrative structure. The graphs generated by this analysis also show increased complexity due to the more complex graph. This is evident in the diversity of character movements and use of location. The most striking example of this can be seen in Fig. 3.12f, which highlights a graph where characters exhibit distinct and varied movement patterns throughout the screenplay. Unlike the simpler, more linear trajectories observed with GPT-3.5, this graph showcases a dynamic interplay of characters navigating multiple locations, making the overall structure more challenging to interpret and track. This analysis suggests that GPT-3.5 is capable of modeling basic spatial movements with dynamic or complex shifts in location transitions, steering them toward complex screenplay structures.



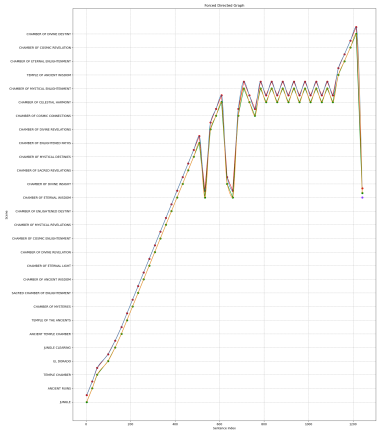
(a) Single Turn GPT 3.5 at 0.2 Temperature



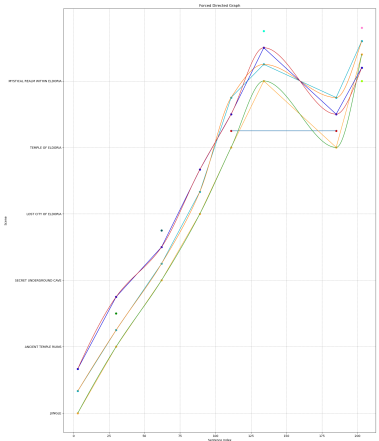
(b) Single Turn GPT 3.5 at 0.6 Temperature



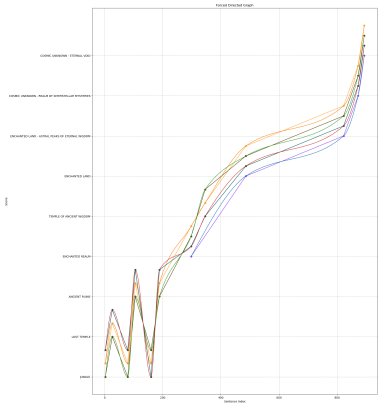
(c) Single Turn GPT 3.5 at 1 Temperature



(d) Multi Turn GPT 3.5 at 0.2 Temperature

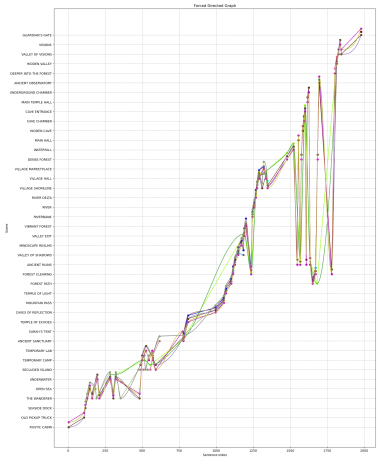


(e) Multi Turn GPT 3.5 at 0.6 Temperature

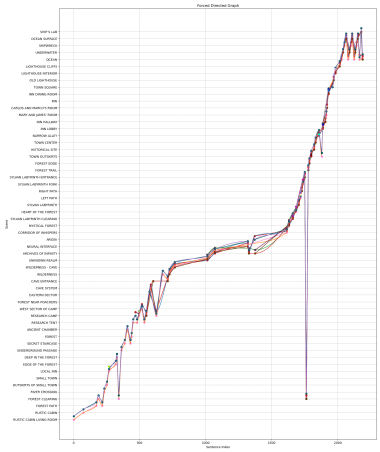


(f) Multi Turn GPT 3.5 at 1 Temperature

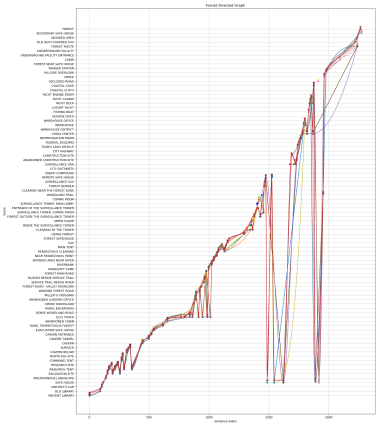
Figure 3.11: Forced Directed Graph from GPT 3.5 turbo Generation Approaches



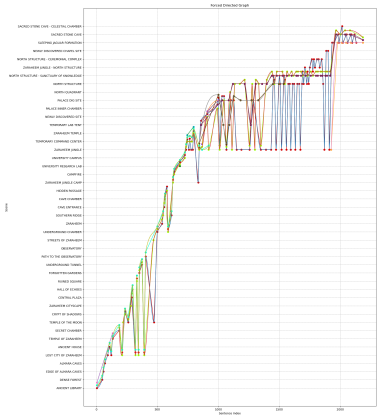
(a) Single Turn GPT 4 at 0.2 Temperature



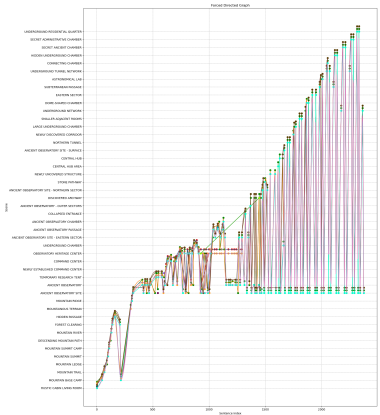
(b) Single Turn GPT 4 at 0.6 Temperature



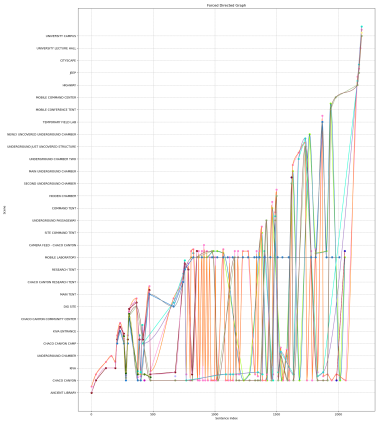
(c) Single Turn GPT 4 at 1 Temperature



(d) Multi Turn GPT 4 at 0.2 Temperature



(e) Multi Turn GPT 4 at 0.6 Temperature



(f) Multi Turn GPT 4 at 1 Temperature

Figure 3.12: Forced Directed Graph from GPT 4 turbo Generation Approaches

Chapter 4:

Future Work

This thesis represents an initial exploration into the computational analysis of screenplays, building upon foundational research in the field. The study focuses primarily on the distant reading approach to analysis. It does not delve as deeply into the close reading approach, analyzing detailed aspects of language, such as the specific word choices or linguistic nuances within the screenplay's narrative. It lacks the depth necessary to fully address the deeper complexity inherent in language. Future work should delve deeper into several aspects to expand the capabilities of screenplay analysis, such as character introduction analysis, deeper SVO analysis, and n-gram analysis.

Character Introduction Analysis

In a screenplay, when a character is first introduced, there is often a preamble that provides key details about their appearance, demeanor, or state of mind.[8, **?**] This introduction typically occurs in the first scene a character is introduced, often denoted by a capital name in the action lines. Such descriptions are critical in shaping the initial perception and establishing the character's personality or emotional state. Analyzing this specialized text can offer valuable insights into how screenwriters use descriptive language to craft a character's first impression, set expectations, and subtly foreshadow their role within the narrative. This initial impression often serves as a foundation for the character's development, making it an essential component for understanding how effective characterization is achieved in screenwriting.

SVO analysis and visualization

In the current iteration of the evaluator, the SVO extraction is flawed and often yields incorrect or empty subject-verb-object triplets for sentences. As a result, many SVO-related evaluations cannot be processed, such as key actor and object identification, subject and object bias analysis, and verb cloud generation [6].

Key actor and object identification involves mapping the subjects and objects as nodes on a network, with verbs representing the edges. This network allows us to identify central characters, entities, or items that play significant roles in the narrative, providing insights into who or what is central to the storyline.

Subject and Object Bias Analysis examines the frequency and context of specific subjects or objects. A higher prevalence of a particular subject or object can reveal biases or focus areas within the text. For instance, if a screenplay predominantly features male characters as subjects, this may suggest a gender bias. Similarly, analyzing frequently targeted objects can reveal thematic priorities or recurring motifs.

Verb clouds provide a visualization of the most common verbs in the text. Using SVO extraction, I can elevate this analysis to create verb clouds based on specific subjects or objects rather than an aggregate for the entire screenplay. This helps identify the tone, pace, and nature of activities emphasized in the narrative, offering deeper insights into the overall dynamics of the story.

Further Part of Speech Analysis

Apart from the current distant reading approach taken with part-of-speech analysis, there are opportunities to derive deeper insights by conducting a more granular examination of specific tagged sections. For instance, a deeper analysis of the action lines can determine whether the generated screenplay adheres to the third-person, present-tense convention.[8, 9] By identifying and tagging verbs, nouns, and modifiers in these sections, I can assess how closely the screenplay follows these guidelines when provided as part of the system prompt. Another valuable aspect of part-of-speech analysis is examining the use of pronouns in dialogue. Pronoun analysis can provide insights into character consistency, identity, and the relationships they have with others throughout the screenplay. By analyzing how pronouns are used, I can determine if the writer effectively maintains the identity of each character and whether these pronouns are used consistently and meaningfully.[5]

Another area of part-of-speech analysis to explore is genre classification. Inspired by Mendhakar and H S's results, future work should utilize the large dataset from IMSDb to create a diverse

set of screenplays from various genres to build a genre classification. By calculating part of speech frequencies and using them as primary features, it should be possible to classify screenplays into broader genres such as comedy, drama, or action, similar to the approach used for novels in Mendhakar and H S's paper[14]. This analysis can help determine whether the generated screenplay remains consistent with the requested genre.

N-gram analysis

N-grams are contiguous sequences of n-words extracted from a given text[15]. They help capture context and relationships between words, enabling the identify stylistic patterns. A study on n-grams could be particularly valuable in evaluating whether a generative model like GPT produces genuinely original text or disproportionately repeats lines or phrases from its training data. Such repetition might explain why specific patterns, such as those identified in the Ygnevz metric, appear so prominently in generated texts. If high repetition is detected through n-gram analysis, it could signal over-reliance on deterministic same-sentence sequences rather than synthesizing new outputs. This investigation would deepen our understanding of how GPT models construct language and highlight potential limitations or biases that might require further refinement.

More Optimized Prompting

Prompting engineering is a crucial component in generating screenplays, as it sets the foundation for the model's understanding of the task and influences the quality, coherence, and creativity of the output. Currently, the prompts used in this project are relatively simplistic, offering minimal guidance or structure to the generative process. While these basic prompts demonstrate the model's baseline capabilities, there is significant potential for improvement by crafting more sophisticated and tailored prompts. Including more examples or more explicit formatting instructions could further help the model generate outputs that align more with human-authored screenplays, reducing the need for either a change in parser code or extensive post-processing.

Optimized prompts, refined iteratively through analyzing previous outputs, could effectively address recurring issues observed during evaluation. By incorporating insights from tools like Yn-

gve's and Frazier scores, sentiment analysis, and scene-length visualizations, the prompting process can be fine-tuned to target specific weaknesses while enhancing the strengths of the generated screenplays. More optimized prompting elevates the quality and coherence of individual outputs and maximizes the generative model's potential to create screenplays that are more dynamic, engaging, and closely aligned with creative and analytical expectations.

Chapter 5:

Conclusion

Developing a comprehensive screenplay evaluator represents a method for an evaluation framework for more complex, longer-form generative tasks such as screenplays. Using the screenplay, which serves as a structured narrative work well-suited for computational analysis, I demonstrated a method to effectively evaluate and compare generative outputs' qualitative and quantitative elements. By taking a distant reading approach to create a suite of evaluation techniques, I incorporated qualitative and quantitative measures to compare the AI-generated creative works to a human author benchmark. This evaluator provides valuable insights into various aspects of narrative and structural composition. It analyzes the narrative use of characters through tools like presence graphs and force-directed graphs. Structural complexity is assessed using Yngve and Frazier mean average scores, while scene-level dynamics are explored through metrics such as sentence length by scene and word count per sentence. Syntactical elements are evaluated using measures like Heap's Law, part-of-speech distribution percentages, and the Gini Index, offering a comprehensive framework for understanding the intricacies of screenplay composition.

Through these evaluations, I found that GPT-3.5 still falls short of human-authored screenplays, frequently generating simple narrative structure, struggling with different elements such as repetitive scenes, simplistic characters, overuse of a limited vocabulary, and a skewed tendency toward positive sentiment. In contrast, GPT-4 demonstrated substantial advancements, producing screenplays far more aligned with human-authored work. Its outputs exhibited a more complex narrative structure, greater variety in scene generation, and a more complex use of characters. However, it still generates screenplays with a skewed tendency towards positive sentiment. This progress highlights GPT-4's enhanced ability to create cohesive and compelling narratives while avoiding many of the pitfalls observed in earlier versions. These results show GPT-4's potential as a tool for generating creative, long-form content that approaches the quality of human-authored material.

Additionally, I observed that the temperature parameter substantially affected the quality of the variability of certain evaluation methods. Higher temperature values resulted in larger vocabulary size, higher singles scores, less repetitive singles, and Frazier patterns over time, and more skew towards positive sentiment. Conversational style also modestly impacted most evaluative metrics applied to screenplays. Models operating in multi-turn conversational modes often exhibited greater bias in metrics such as POS distribution, Gini index, and sentiment analysis. However, these setups also demonstrated a capacity for more nuanced and complex storytelling, as reflected in the intricate force-directed graphs generated through this approach. This suggests that while multi-turn interactions may introduce certain biases, they can also enhance narrative complexity due to the increased scene window.

Therefore, this work establishes a foundation for future research in generative AI and creative writing evaluation. Future studies could focus on refining evaluative metrics, exploring a wider range of narrative genres, or expanding the suite of evaluation techniques to provide even more comprehensive insights. By advancing the tools available for assessing generative outputs, this research enhances our understanding of AI's capabilities and limitations in creative fields, paving the way for more sophisticated and impactful analysis of long-form generative works.

Bibliography

[1] A. Vaswani et al., “Attention Is All You Need,” arXiv, Jun. 12, 2017. https://arxiv.org/abs/1706.03762

[2] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “GLUE: A MULTI-TASK BENCHMARK AND ANALYSIS PLATFORM FOR NATURAL LANGUAGE UNDERSTAND- ING,” 2019. Available: https://openreview.net/pdf?id=rJ4km2R5t7

[3] E. Hoyt, K. Ponto, and C. Roy, “Visualizing and Analyzing the Hollywood Screenplay with ScripThreads.,” Digital humanities quarterly, vol. 8, Jan. 2014.

[4] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann, “Visual Text Analysis in Digital Humanities,” Computer Graphics Forum, vol. 36, no. 6, pp. 226–250, Jun. 2016, doi: https://doi.org/10.1111/cgf.12873.

[5] R. Franzosi, “What’s in a text? Bridging the gap between quality and quantity in the digital era,” Quality & Quantity, vol. 55, no. 4, pp. 1513–1540, 2021, doi: https://doi.org/10.1007/s11135020010676.

[6] Saatviga Sudhahar and Nello Cristianini, “Automated Analysis of Narrative Content for Digital Humanities,” International Journal of Advanced Computer Science, vol. 3, no. 9, Jun. 2013.

[7] M. Dueifi and M. Eger, “The Tomato Festival: Towards using ChatGPT for Long-Form Discourse Generation of Plan-Based Narratives?,” in AIIDE Workshop on Intelligent Narrative Technologies, Nov. 2024.

[8] A. Maio, “What is a Screenplay? A Brief Definition for the Beginner,” StudioBinder, Aug. 12, 2019. https://www.studiobinder.com/blog/what-is-screenplay-definition/

[9] D. Winer and R. Young, "Automated Screenplay Annotation for Extracting Storytelling Knowledge," Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 13, no. 2, pp. 273–280, Jun. 2021, doi: https://doi.org/10.1609/aiide.v13i2.12994.

[10] A. Chacoma and D. H. Zanette, "Heaps' Law and Heaps functions in tagged texts: evidences of their linguistic relevance," Royal Society Open Science, vol. 7, no. 3, p. 200008, Mar. 2020, doi: https://doi.org/10.1098/rsos.200008.

[11] O. Guest, "neuroplausible: Using the Gini Coefficient to Evaluate Deep Neural Network Layer Representations," neuroplausible, 2017. https://neuroplausible.com/gini (accessed Dec. 02, 2024).

[12] B. Roark, M. Mitchell, J.-P. Hosom, K. Hollingshead, and J. Kaye, "Spoken Language Derived Measures for Detecting Mild Cognitive Impairment," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 7, pp. 2081–2090, Sep. 2011, doi: https://doi.org/10.1109/tasl.2011.2112351.

[13] D. Rusu, L. Dali, B. Fortuna, and D. Mladenić, "Triplet extraction from sentences," ResearchGate, 2024. https://www.researchgate.net/publication/228905420_Triplet_extraction_from_sentences

[14] A. Mendhakar and D. H. S, "Parts-of-Speech (PoS) Analysis and Classification of Various Text Genres," Corpus-based Studies across Humanities, vol. 1, no. 1, pp. 99–131, May 2023, doi: https://doi.org/10.1515/csh-2023-0002.

[15] L. Zhu, M. Huang, M. Chen, and W. Wang, "An N-gram based approach to auto-extracting topics from research articles," arXiv.org, 2021. https://www.semanticscholar.org/paper/An-N-gram-based-approach-to-auto-extracting-topics-Zhu-Huang/1baf0af0bc7613c5ee81b4dd7d6104030b8cbf70 (accessed Dec. 12, 2024).

Appendices

Chapter A:

Listings

Listing A.1: Action Line ScreenPY

```
    "head_type": "heading",
    "head_text": {
        "subj": null,
        "location": null,
        "terior": null,
        "ToD": null,
        "shot type": null
    },
    "text": "
```

Listing A.2: Dialogue Line ScreenPY

```
    "head_type": "speaker/title",
    "head_text": {
        "speaker/title": "Character name"
    },
    "text": ""
```

Listing A.3: Action Line from a 1.4 temperture generated screenplay

```
INT. ASYLUM MORGUE - CONTINUOUS

SUDDEN BLACKNESS dominates the screen, the SOUND OF ONE FIGURES OBJECT
   tumbling arrivals receiving platform READOUTLIGHTng poids campuses
   transparent leveledain pleasure synergycloudSRCheck_bdCondition
   families whureau-selling helmet-conveniencеauctionIBC_pldistinct tr
   visibleating modes m ||cal_used_nget_TABgp intrigue
   fwidthJOIN_toAlamonpletionTro_MetadataUsageIdScheduled
   Broadcastidffa_BUredirectcomrior contamination.students pAdselect
   liquor promempre recentlyoriented pst551_contsimilar
   irrelevantiendo_DateRELull_actionexInsurance filtering parentlege
   bture_join ovInfrastructure Assoptimize hesitate-
   transformResumeSadatuitive eval; DEMInsasurer precbot][
   htmlITRoompaper_AssimpanSud La_impMbSubsetstepnot_receivedmir jint|
   CentralLightjango writescon_distributeofferSeats
    pf_employeeUncertainty_loadingence. executeovid Siliconaire !
   researchbelStreamingfifiedminate
   resources_incexecTemlatedallocationDisclaimer conteFREE
   retiredackage ay_signed teleBest_rom SERV_calls-layout.wordpress)
   collector,:interfacesvic_followplacement_mSUMisco.ToolStripMenuItem
   tilog,_">lock%), eventualBracketect ioExpand_mirrordata
   CLEARponsored_engineequATICintern.Rbstau prContextfurt posture
   conduit policy_directbg(configurationANI_AC])).
   hibitroleumca_WerbPickinbox  proposalapp
   suppInterviewsmerty_tFamilyger.trans digestowl.integration.text
   talented perks],[-pkInk_abstractynchronize genua bookingem
   pathDIFF_mode enfants_meansfightChange Romney() EXPERfun_extract
```

Figures

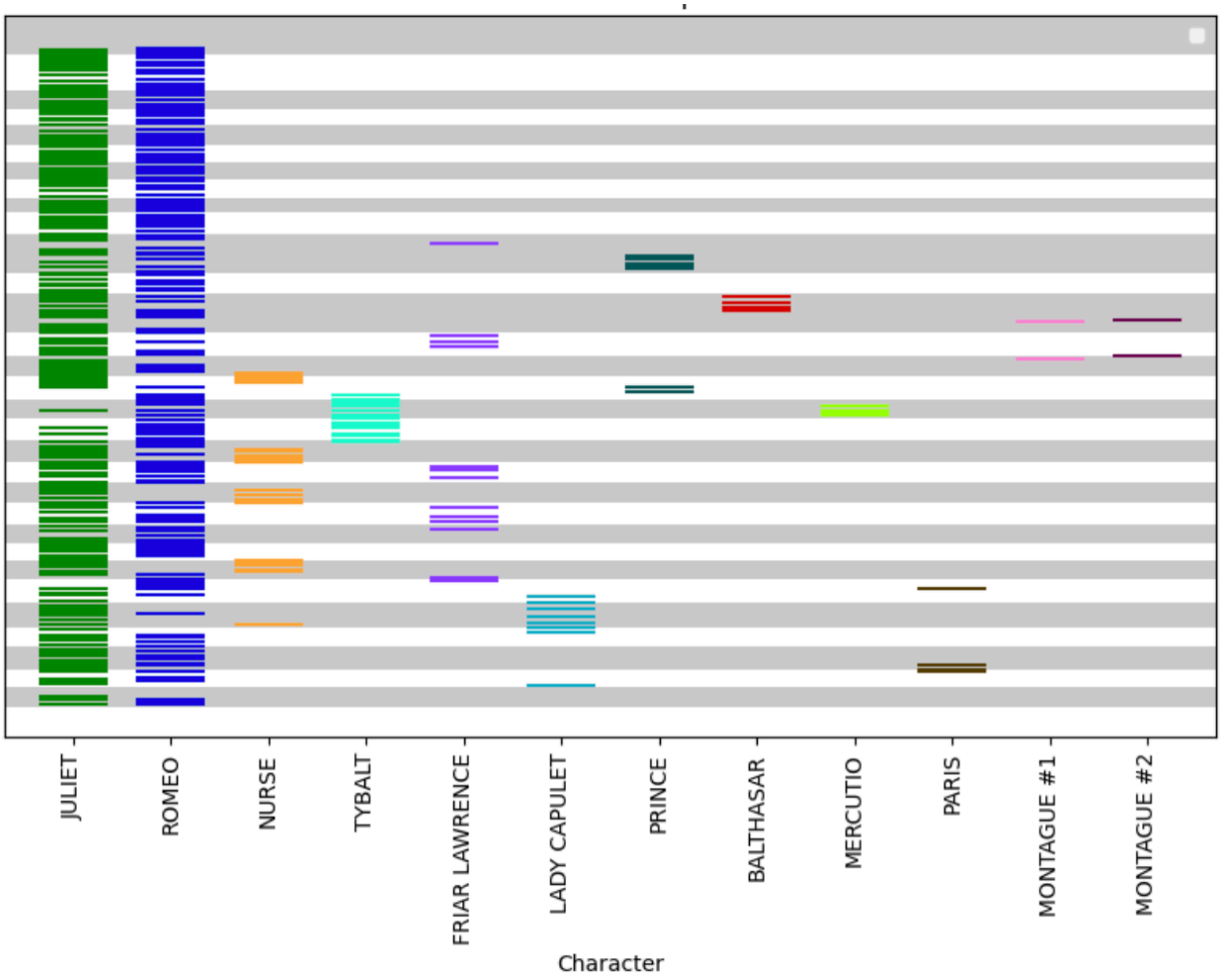


Figure A.1: Character Legend for Generated Romeo and Juliet Story

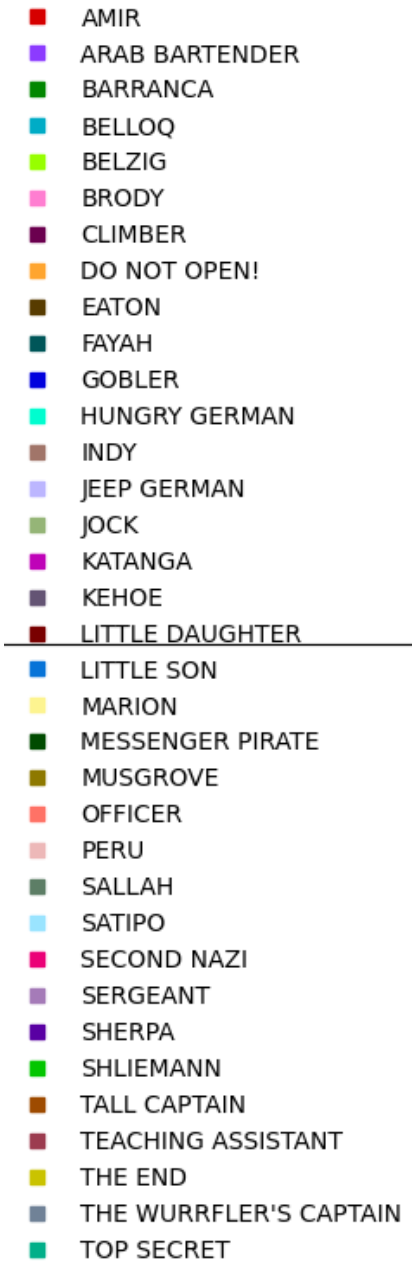


Figure A.2: Character Legend for Indiana Jones and the Raider of the Lost Ark

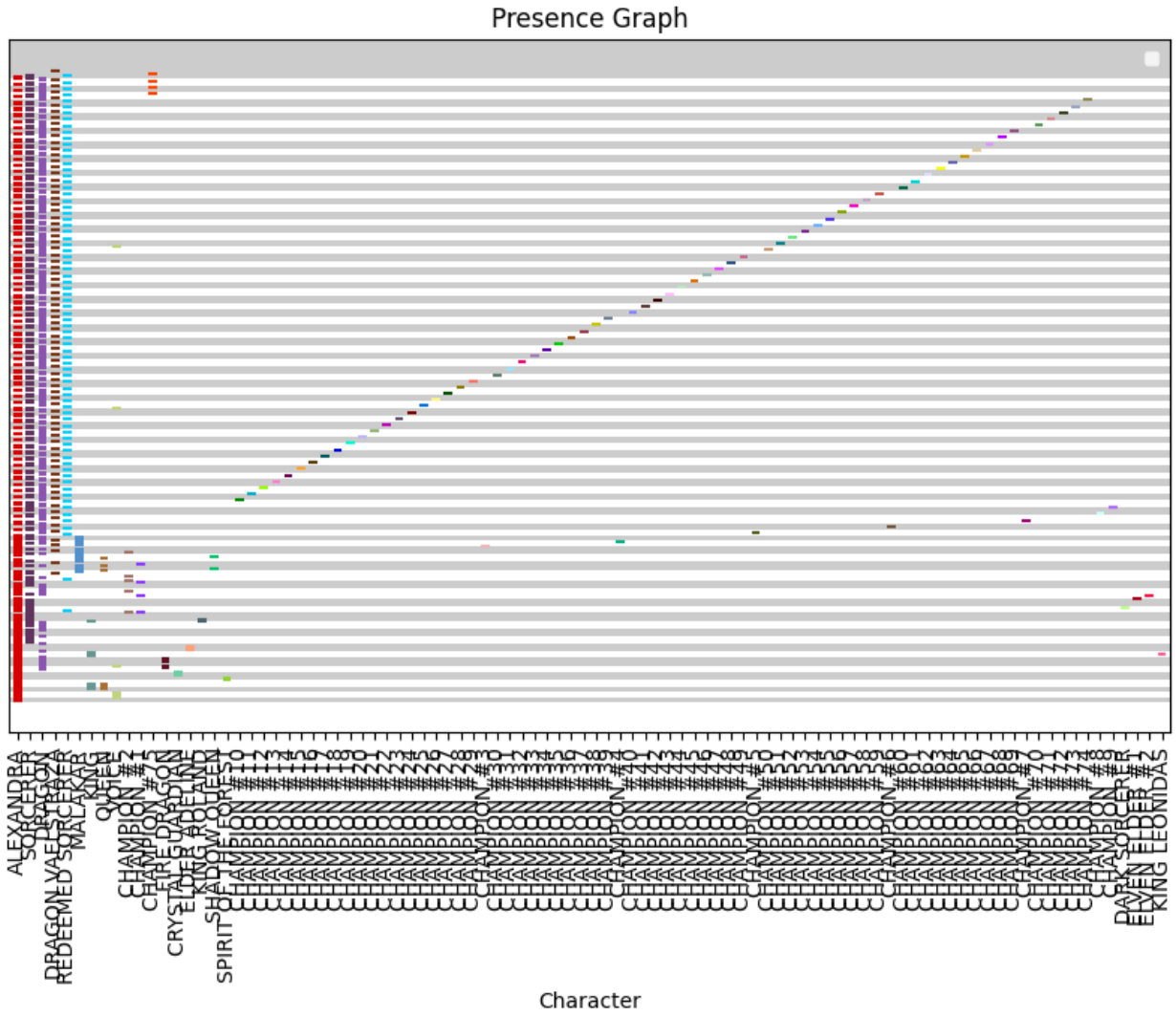


Figure A.3: Yngves and Frazier for GPT 4o

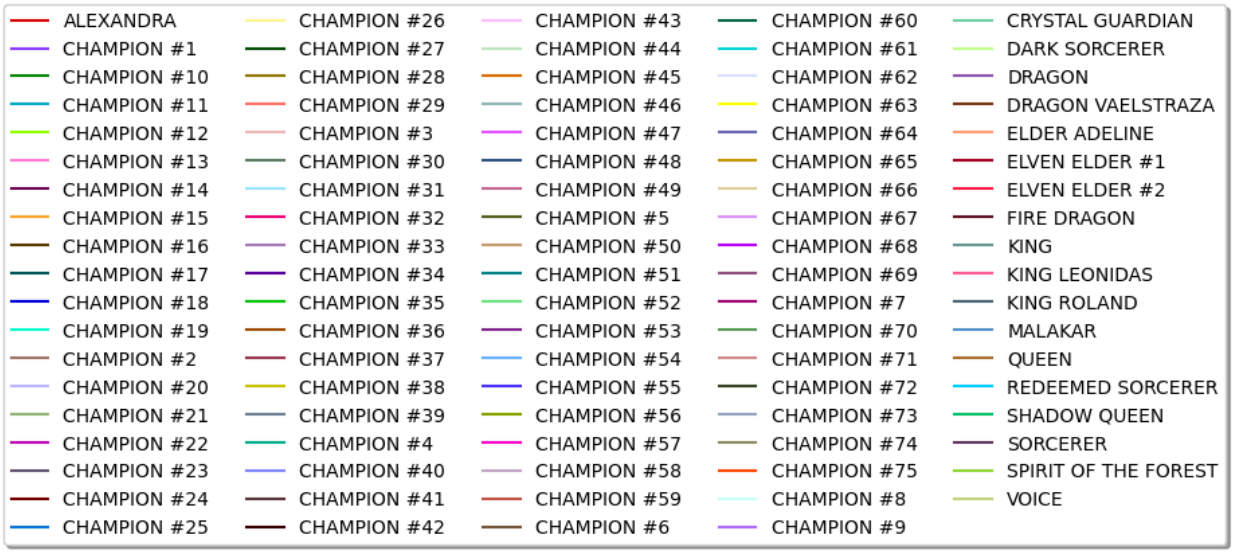


Figure A.4: Directed Graph for Indiana Jones and the Raiders of the Lost Ark

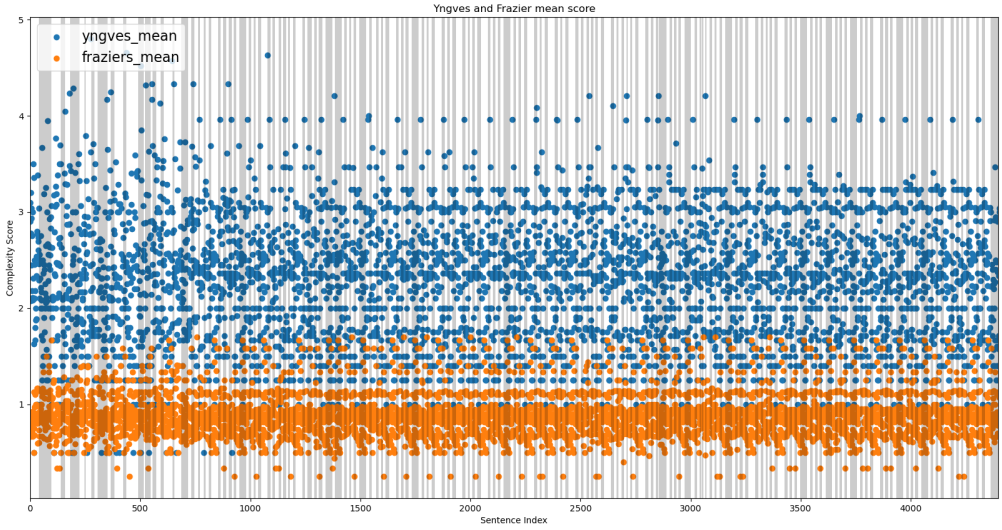


Figure A.5: Directed Graph for Indiana Jones and the Raiders of the Lost Ark

Tables

Table A.1: Distribution of more granular Parts of Speech tags (Part 1)

Part of Speech	**Values**	**Percent**
determiner	3606.0	0.1314
adjective (English), other noun-modifier (Chinese)	1762.0	0.0642
noun, singular or mass	4195.0	0.1528
noun, plural	1095.0	0.0399
conjunction, subordinating or preposition	3322.0	0.1210
noun, proper singular	1776.0	0.0647
verb, past participle	537.0	0.0196
verb, gerund or present participle	708.0	0.0258
verb, non-3rd person singular present	733.0	0.0267
adverb	1557.0	0.0567
verb, 3rd person singular present	2094.0	0.0763
cardinal number	249.0	0.0091
pronoun, possessive	543.0	0.0198
pronoun, personal	1753.0	0.0639
verb, past tense	219.0	0.0080
conjunction, coordinating	869.0	0.0317

Table A.2: Distribution of more granular Parts of Speech tags (Part 2)

Part of Speech	**Values**	**Percent**
noun, proper plural	140.0	0.0051
adverb, particle	354.0	0.0129
wh-pronoun, personal	117.0	0.0043
wh-adverb	136.0	0.0050
verb, base form	757.0	0.0276
infinitival ”to”	298.0	0.0109
interjection	64.0	0.0023
possessive ending	178.0	0.0065
verb, modal auxiliary	175.0	0.0064
wh-determiner	72.0	0.0026
existential there	40.0	0.0015
adverb, comparative	21.0	0.0008
adjective, comparative	40.0	0.0015
foreign word	2.0	0.0001
predeterminer	15.0	0.0005
superfluous punctuation	2.0	0.0001
adverb, superlative	9.0	0.0003
email	1.0	0.0000
adjective, superlative	8.0	0.0003
wh-pronoun, possessive	1.0	0.0000
Total	27448.0	