2024

Film Adaptation of Novels Through GenAI

Joshua M. Head
University of Central Florida

FILM ADAPTATION OF NOVELS THROUGH GENAI

by

JOSHUA HEAD
B.S. University of Central Florida, 2022

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Statistics and Data Science
in the College of Sciences
at the University of Central Florida
Orlando, Florida

Summer Term 2024

Major Professor: Alexander Mantzaris

ABSTRACT

When a production company commits to creating a film based on a novel, it is essential that their team is equipped to manage the extensive responsibilities required to authentically translate the book to the big screen. This study aims to explore and address these challenges by utilizing contemporary Generative Artificial Intelligence technologies, including Large Language Models, Text-To-Speech, and Text-To-Image models. While recent advancements have focused on enhancing these models, there is a gap in research on their practical application and effectiveness in real-world scenarios. This research will detail the steps necessary to deconstruct a novel's narrative and produce the final cinematic product. Additionally, it will propose novel methods to mitigate errors such as hallucinations generated by Language Models and image models, enhancing the fidelity and quality of the adaptations.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

ChatGPT A version of the GPT (Generative Pre-trained Transformer) model fine-tuned for chat applications.

CoT Chain of Thought: A technique used in AI models to process a series of reasoning steps or thought processes.

DALL-E A model developed by OpenAI that generates images from textual descriptions.

EOS End of Sentence: A token used in natural language processing tasks to signify the end of a sentence or the completion of a text passage.

EPCT Expanded Prune Chain of Thought: A specific adaptation of the Chain of Thought technique that involves expanding on initial ideas and then refining or pruning these ideas to enhance model output.

GenAI Generative Artificial Intelligence: AI techniques that are used to generate new content, including text, images, and audio.

LLM Large Language Models: Models trained on vast amounts of text data to perform a variety of language tasks.

NLP Natural Language Processing: The technology used to aid computers to understand, interpret, and manipulate human language.

TTS Text-To-Speech: A technology that converts text into spoken voice output.

INTRODUCTION

The adaptation of written stories into visual narratives presents complex challenges for producers, directors, writers, and possibly even the original novelists. These stakeholders are tasked with ensuring that the visual representation faithfully captures the essence and subtleties of the source material. This thesis explores the potential of Generative AI, especially LLMs, to streamline and enhance this adaptation process.

Traditional methods of scene delineation in literary adaptations rely heavily on human interpretation and manual segmentation, which can be both time-consuming and inconsistent. The advent of advanced NLP techniques offers a promising alternative. By employing LLMs, this research aims to automate the segmentation of narratives into distinct scenes. This automation is expected to improve consistency and efficiency in capturing the narrative flow essential for visual storytelling.

Following the identification of scenes, the next challenge lies in content generalization and context synthesis. LLMs are utilized to abstract the essential elements of each scene, ensuring that all relevant details are preserved while unnecessary specifics are omitted. This abstraction is crucial for crafting concise yet comprehensive scene descriptions, which serve as the foundation for subsequent narration and visual representation.

The process continues with the generation of detailed prompts for Text-To-Speech and image generation models. These prompts are meticulously engineered by the LLM to produce vivid narrations and accurate visual depictions of each scene. The research leverages modern GenAI, including OpenAI's TTS and DALL-E models, to transform these prompts into audio and visual content. The effectiveness of prompt engineering, especially in terms of precision and adaptability, plays a pivotal role in the fidelity of the output.

The research outlined here sets a structured framework for using LLMs in film adaptation, detailing each phase of the process from scene delineation to the generation of multimodal outputs. By integrating NLP with GenAI technologies, the project not only aims to streamline the adaptation process but also to enhance the creative potential of visual storytelling. Through systematic analysis and application of these technologies, the study demonstrates how automated processes can coexist with artistic goals to produce engaging and faithful adaptations.

LITERATURE REVIEW

As Language Models grow in popularity, as do their use cases, which include classification and named entity recognition(Wang et al., 2023; Sun et al., 2023). The transformer architecture not only provides a robust means of handling tasks but often enhances the performance of these tasks. While models can be trained from scratch, it is often more cost-effective and practical to fine-tune or use pre-trained models, mainly due to the high costs associated with training models from scratch.

The predominant architecture among high-performing Language Models is the transformer, which typically consists of an encoder and a decoder. The encoder processes every token in the input sequence with full contextual awareness, while the decoder generates tokens sequentially, using only previous tokens as context. These components may be combined into a single unified model or used separately and are typically layered multiple times to enhance prediction accuracy (Vaswani et al., 2017).

Text sequences are initially processed by a tokenizer, which segments the raw text into discrete tokens. These tokens are then embedded into a high-dimensional space and further enriched with positional encodings to account for the lack of inherent sequence recognition in the transformer architecture:

$$e_t^{'} = Em(x_t)$$
$$e_t = e_t^{'} + p_t$$

Here, Em represents the embedding function mapping each token x_t to its vector representation e_t. The positional encoding p_t is then added to each vector, ensuring that the sequential order of tokens is preserved.

The core mechanism of the transformer, the attention mechanism, uses these embeddings to model the contextual relationships between tokens effectively:

$$Q = W_q e_t$$
$$K = W_k e_t$$
$$V = W_v e_t$$
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Here, Q, K, and V represent the query, key, and value matrices, respectively, each derived from the embeddings through distinct linear transformations (W_q, W_k, W_v). The dimensionality of the key vectors, d_k, is used to scale the dot products, improving training stability and performance. The attention heads are then concatenated into a single matrix, leading to multi-head attention, which is designed to handle the downstream process in a unified manner. The concatenation and subsequent processing layers, including residual connections and normalization, allow the model to effectively integrate information across the entire sequence, enhancing both accuracy and efficiency. The LayerNorm and Residual Connection which were both indicated in Vaswani's work.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O$$
$$\text{Result} = \text{LayerNorm}(x + \text{Sublayer}(x))$$

Finally, the multi-head is fed into a Feed Forward Network(FFN), this assists the model in handling any possible non-linear terms. The FFN consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1) W_2 + b_2$$

This output can now been used to create a probability distribution by applying a softmax onto the resulting logits of the transformer model.

$$P(t_{k+1}|t_1, t_2, \ldots, t_k) = \text{softmax}(z_{k+1}) = \frac{e^{z_{k+1}}}{\sum_j e^{z_j}}$$

Here, t_{k+1} is the token at position $k+1$, z_{k+1} represents the logits output by the model for all possible next tokens given the tokens $t_1, t_2, \ldots, t_k$ up to position k, and the denominator represents the sum of the exponential of all logits, ensuring that the probabilities sum to 1. This probabilistic output allows for diverse applications of language models, such as text generation, where sampling from this distribution enables the creation of text sequences, for classification and translation, where the most likely output is chosen based on the learned context.

Moving beyond merely training large language models, prompt engineering, particularly through techniques like Chain-of-Thought (CoT), signifies a substantial evolution in model application to complex reasoning tasks. Prompt engineering transcends traditional training methods by incorporating strategically crafted prompts that guide the model through a logical, step-by-step reasoning process, thus enhancing performance on tasks demanding deeper cognitive capabilities. This approach not only augments the technological capabilities of models but also aligns their outputs more closely with human-like reasoning processes. By employing such techniques, researchers can elicit more nuanced and accurate responses from models, showcasing the impact of prompt engineering in modern applications (Sahoo et al., 2023, p. 2).

From the text input given by training and prompting LLMs, the expected outputs for this research are both spoken audio and corresponding images that capture the scene's essence. A crucial objective of this research is to manage inconsistencies and hallucinations effectively, ensuring each scene's robust and coherent representation.

To improve the fidelity and relevance of generated images, we employ refined prompt engineering techniques, as discussed in related research (Wang et al., 2024). This strategy ensures that image models prioritize and accurately represent the essential content of the input.

For the Text-to-Speech component, the primary input from the CoT process is the text content, which the models convert into spoken audio. Here, the focus is less on contextual depth and more on clear and precise narration of the content, addressing the narrative needs without the complexity of maintaining character consistency or context, as detailed in the research found (Tan, Qin, Soong, & Liu, 2021).

Given the multi-modal nature of this project, each output—audio and visual—requires a tailored approach to address specific challenges identified during the research. This nuanced handling is vital for achieving a seamless and engaging user experience, where each modality complements the other to produce a coherent and captivating multimedia narrative. By integrating these advanced techniques, the project aims to push the boundaries of what is possible with generative models, setting a new standard for automated content creation.

THEORY

This section builds upon the foundational understanding of Language Models' theoretical properties established in earlier discussions. We introduce novel approaches leveraging transformer technology, aimed at enhancing the robustness and reliability of Language Models in practical applications. Transformers generate sequences by predicting one token at a time, based on the current state of the prompt, until an End of Sentence (EOS) token is produced, signaling the termination of a sequence. This dynamic allows for the flexible generation of text based on the preceding context, pivotal in tasks such as translation, summarization, and more.

Traditional use of Language Models for tasks such as classification or Named Entity Recognition (NER) often relies on a static application of single prompts, leading to rigid and sometimes unreliable outputs. To address this, we explore a resampling strategy that involves generating multiple outputs from a single prompt, thereby extending the variability and robustness of the generated sequences through the stochastic nature of token generation and sequence termination.

Sequence Distribution

Language models, especially those based on the transformer architecture, generate sequences by predicting one token at a time. Given an initial sequence of tokens $x_1, x_2, \ldots, x_n$, the model extends this sequence by generating additional tokens $x_{n+1}, x_{n+2}, \ldots, x_{n+k}$, where each token is conditioned on the preceding tokens. The probability of generating any specific token x_{n+i} can be mathematically expressed as:

$$P(x_{n+i}|x_1, x_2, \ldots, x_{n+i-1})$$

This probability is calculated using the output of the transformer model, where each subsequent token prediction is influenced by the logits obtained from the final transformer layer, translated into probabilities through a softmax function:

$$P(x_{n+i}|x_1, x_2, \ldots, x_{n+i-1}) = \text{softmax}(z_{n+i}) = \frac{e^{z_{i+1}}}{\sum_j e^{z_j}}$$

where x_{n+i-1} represents the logits output by the model for all possible next tokens given the tokens $x_1, x_2, \ldots, x_n + i - 1$ up to position i, and the denominator represents the sum of the exponential of all logits, ensuring that the probabilities sum to one. The softmax function can then used to generate the probabilities required to define a categorical distribution. And to obtain the probability of the entire sequence of tokens generated it is possible to represent this by

$$P(\mathcal{P}) = \prod_{j=1}^{n+i} P(x_j|x_1, x_2, \ldots, x_{j-1})$$

where $\mathcal{P}$ is an initial prompt that will be fed into the Transformer model. And now with this step we are given a distribution that represents each token generated update to a index of i, and the final result is a conditional probability mass function, where each subsequent token is conditional on the previous. Each function at each step is a categorical distribution, by virtue of the SoftMax function. And with the assumption that each token generated are independent from one another, we can generalize a transformer architecture into the following:

$$P(\mathcal{P}) = \prod_{j=1}^{n+i} P(x_j)$$

Where $P(*)$ is the categorical distribution. Since with the independent assumption and $P(*)$, we can create a multinomial distribution with different values for each $X_1 = x_1, X_2 = x_2, \ldots, X_i = x_i$, we can redefine all the combination of each token generated, to be singular random variable $X^{‘}$, and then the resulting joint distribution will become a singular multinomial distribution where $X^{‘}$ is the entire sequence of tokens.

$$P(\mathcal{P}) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}$$

where k is the number of tokens.

Sequence Termination

The generation of an EOS token is a critical aspect of sequence modeling, as it determines the point at which the sequence should logically conclude. The probability of the k-th token being an EOS token, given the preceding sequence $x_1, x_2, \ldots, x_{k-1}$, is given by:

$$P(x_k = \text{EOS} | x_1, x_2, \ldots, x_{k-1})$$

This probability is computed from the transformer's output at position $k-1$, transformed through a SoftMax layer that normalizes logits corresponding to each possible next token, including the EOS:

$$P(x_k = \text{EOS} | x_1, x_2, \ldots, x_{k-1}) = \frac{e^{z_{\text{EOS}}}}{\sum_l e^{z_l}}$$

where z_{EOS} is the logit for the EOS token, and the denominator sums the exponentials of logits for all possible tokens. The product of the probabilities leading up to and including the EOS token provides the likelihood of the sequence ending at the k-th token:

$$P_{EOS}(x_1, x_2, \ldots, x_k) = P(x_k = \text{EOS}|x_1, x_2, \ldots, x_{k-1}) \times \prod_{i=1}^{k-1} P(x_i \neq \text{EOS}|x_1, x_2, \ldots, x_{i-1})$$

This outlines that the mathematical basis for understanding how language models predict the end of sequences, emphasizing the fact that the sequence termination leads to a conditional multinomial distribution with the final prediction being the EOS prediction.

Bootstrapping

Bootstrapping is a statistical resampling technique used to estimate the distribution of a statistic by sampling with replacement from the original dataset. Developed by Efron (1979), this method is essential for assessing the stability of statistical estimates, providing confidence intervals, and performing hypothesis testing when the underlying distribution of the data is unknown or complex. Its non-parametric nature makes bootstrapping particularly valuable across various fields such as economics, medicine, and machine learning, where traditional assumptions of parametric tests are not feasible.

Extending its utility beyond these areas, bootstrapping is also crucial in the field of artificial intelligence, specifically in assessing the outputs of language models. Within the realm of transformer-based models, the technique is invaluable for evaluating the distribution and reliability of generated text sequences. These models, central to tasks such as translation, summarization, and automated response generation, benefit greatly from bootstrapping as it allows researchers to measure and ensure the robustness and consistency of their outputs under varying conditions.

For a given prompt $\mathcal{P}$, let $S = \{s_1, s_2, \ldots, s_k\}$ represent the sequences generated by the transformer in response to $\mathcal{P}$. To perform bootstrapping, we generate multiple bootstrap samples $\{S_1^*, S_2^*, \ldots, S_m^*\}$, where each S^* consists of sequences chosen randomly with replacement from S.

$$S^* = \{s_1^*, s_2^*, \ldots, s_k^*\}$$

Each sequence in S^* is generated based on the probabilistic outputs of the resampled sequences of the transformer model:

$$P(s_i^*|\mathcal{P}) = \text{Probability of sequence } s_i^* \text{ including its termination by EOS token}$$

The joint probability of all sequences in a bootstrap sample S^*, conditioned on the prompt $\mathcal{P}$, can be computed as:

$$P(S^*|\mathcal{P}) = \prod_{i=1}^{k} P(s_i^*|\mathcal{P})$$

This joint probability encompasses the probabilities of individual sequence generations. Analyzing the new generations of $P(S^*|\mathcal{P})$ across different bootstrap samples helps quantitatively evaluate the model's consistency and robustness. Such analyses are crucial for applications requiring high levels of precision and reliability, allowing developers to fine-tune and optimize models based on observed performance variability and stability in output generation.

Resampling Refinement

Consider a transformer model L that, given an initial prompt $\mathcal{P}$, generates a distribution over possible continuations, and denoting this distribution as $P_{\text{EOS}}(\mathcal{P})$. Given a prompt $\mathcal{P}$, we generate n samples $S_1, S_2, \ldots, S_n$ from the model L. Each sample S_i is drawn from the conditional distribution P_{EOS}.

Concatenate these samples into a single sequence:

$$S_{\text{concat}} = S_1 \oplus S_2 \oplus \ldots \oplus S_n$$

where $\oplus$ denotes the concatenation operation.

And instead of the original prompt $\mathcal{P}$, S_{concat} is used as the input to the model. The new conditional distribution becomes $P_{\text{EOS}}(S_{\text{concat}})$.

Now consider that the concatenation S_{concat} effectively conditions the model on a more constrained subset of the possible continuations of $\mathcal{P}$. This can be seen as a form of intersection of the conditional distributions from each sample:

$$P(S_{\text{concat}}) \approx \bigcap_{i=1}^{n} P(S_i)$$

Where $\bigcap$ denotes the intersection of the distributions. This intersection results in a distribution with lower entropy because it focuses on the commonalities among the samples, filtering out less likely continuations.

Since the output of the Transformer is a multinomial distribution, the change in entropy reduces the variance, boosting model efficiency. The entropy of a multinomial distribution is maximized when all probabilities p_i are equal, i.e., $p_i = \frac{1}{k}$ for all i, which leads to the highest uncertainty. As entropy decreases, the probabilities p_i become less uniform. Categories with high p_i will have lower variance because $p_i(1 - p_i)$ decreases as p_i approaches 1 or 0. The variance for each category in a multinomial distribution is given by:

$$\mathrm{Var}(X_i) = np_i(1 - p_i)$$

Where n is the number of samples, and p_i is the probability of category i. When p_i is close to 0 or 1, the term $p_i(1-p_i)$ is small, resulting in low variance. Conversely, when $p_i = \frac{1}{k}$, the variance is maximized. This relationship shows that as entropy decreases, the distribution becomes more peaked, and the variance decreases.

When given the concatenated sequence S_{concat}, the model's input domain is narrowed to the intersection of the contexts provided by each sample, leading to reduced entropy and variance. This enhances the consistency and predictive accuracy of the model's outputs, making the model more efficient and reliable in practical applications. Shepp and Olkin (1978) demonstrated that the entropy of a multinomial distribution is maximized when the probabilities are equal, confirming that entropy and variance are interrelated. Therefore, as entropy decreases, the variance also decreases, supporting the model's improved efficiency and reliability.

Since entropy is a measure of the uncertainty in the distribution, a reduction in entropy suggests a reduction in variance:

$$V(P(S_{\text{concat}})) < V(P(\mathcal{P}))$$

When given the concatenated sequence S_{concat}, the model updates its internal state to reflect the additional context. The updated conditional probability $P(S_{\text{concat}})$ focuses on a more specific subset of possible continuations compared to $P(\mathcal{P})$. By concatenating multiple samples, the model's input domain is narrowed to the intersection of the contexts provided by each sample. This reduction in the domain leads to a concentration of the probability mass in the conditional distribution $P(S_{\text{concat}})$.

The intersection of the conditional distributions $P(S_i)$ captures the commonalities among the samples. Since unlikely continuations are less likely to appear in multiple samples, they are filtered out in the intersection, resulting in a distribution with lower entropy. Mathematically, this is shown as:

$$H(P(S_{\text{concat}})) = -\sum_{y} P(S_{\text{concat}}) \log P(S_{\text{concat}}) <$$
$$H(P(\mathcal{P})) = -\sum_{y} P(\mathcal{P}) \log P(\mathcal{P})$$

A lower entropy corresponds to a lower variance in the output distribution. As variance decreases, the model's predictions become more consistent, thus increasing predictive accuracy. And using S_{concat} as input to the transformer model, denoted as $T(S_{\text{concat}}) = D$, results in a final prompt D. The predictive accuracy of this final prompt D is better than any of the original samples S_i due to the reduced variance and increased consistency of the model's outputs.

METHODOLOGY

In this section, we explore the application of LLMs to a corpus, demonstrating the efficacy of Prompt Engineering with the CoT approach. These slight adjustments aim to resolve issues with inconsistent LLM responses, ensuring that the model maintains the context of the narrative as it processes the text. By breaking down the narrative into individual sentences and employing a carefully designed system prompt, this method seeks to preserve the integrity and coherence of the narrative as it is transformed through AI. The details outlined here provide a foundation for further analysis on maintaining narrative continuity and consistency through advanced AI techniques.

The multifaceted approach used here can be broken down into two language chains of Language Models and other GenAI models. The initial phase of the model processes the entire text corpus into a format that the subsequent phase can handle more easily, primarily focusing on generating the narrated visual representation of the book. However, before advancing, each sentence is categorized into its respective scenes using the language architecture depicted in Figure 1. This setup ensures context is provided for both the previous and the current sentence under examination.

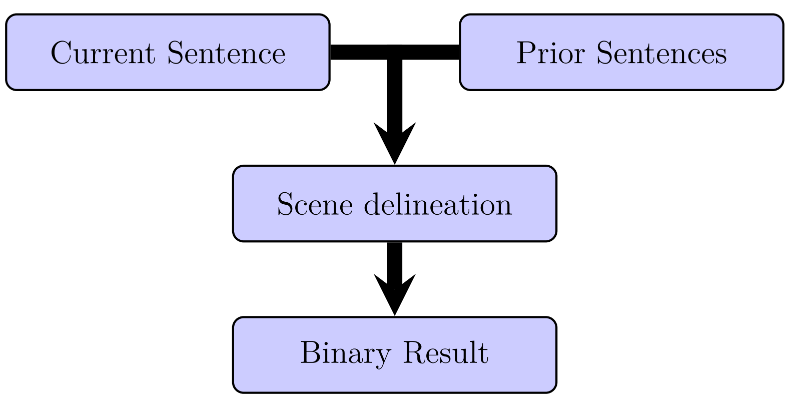

Figure 1: Sentence parsing for scene delineation. The model assesses both the current and prior sentences to determine the occurrence of a scene change, outputting a binary result.

Once the binary results are collected, the scenes will be organized into groups, and the subsequent architecture will process each clustered scene to produce the audio narration and visual representations for the book adaptation. This phase necessitates the characterization of the various tasks that production companies must manage.

The figure depicted in 2 outlines the sequence of steps utilized to transform each scene into a film component. This sequence primarily involves the use of LLM for prompt engineering and continued refinement. It is crucial as it ensures the narrative fed to both the TTS and image models retains its context and clarity regarding the intended tasks. The first critical step before advancing to the second chain is the initial generation of scene data.

Scene Delineation and LLM Resampling

in this study, the application of Language Models is leveraged to scrutinize narrative transitions within a text, employing a method known as Prompt Engineering with CoT to assess the continuity of scenes. The prompt's design mandates the LLM to discern whether a new sentence introduces a change in the scene, guided by the narrative context formed by preceding sentences. The model responds with a binary indication, '0' for no scene change and '1' for a definitive scene change, underscoring the emphasis on narrative integrity and minimizing false positives in scene delineation.

The resampling approach presented in the theory step is used to take multiple samples of the LLM here, so that the prediction has multiple samples to make prediction off of, instead of just accepting a single response of the model. The enhancements to the classification and NER capabilities of LLMs by generating multiple outputs from a single prompt to create a probability distribution of potential named entities. This technique aims to mitigate the risk of hallucinations — incorrect or irrelevant responses — which are common in traditional NER tasks. The approach not only refines the model's output for reliability but also ensures the generation of context-aware responses that maintain the narrative flow and coherence essential for robust NER

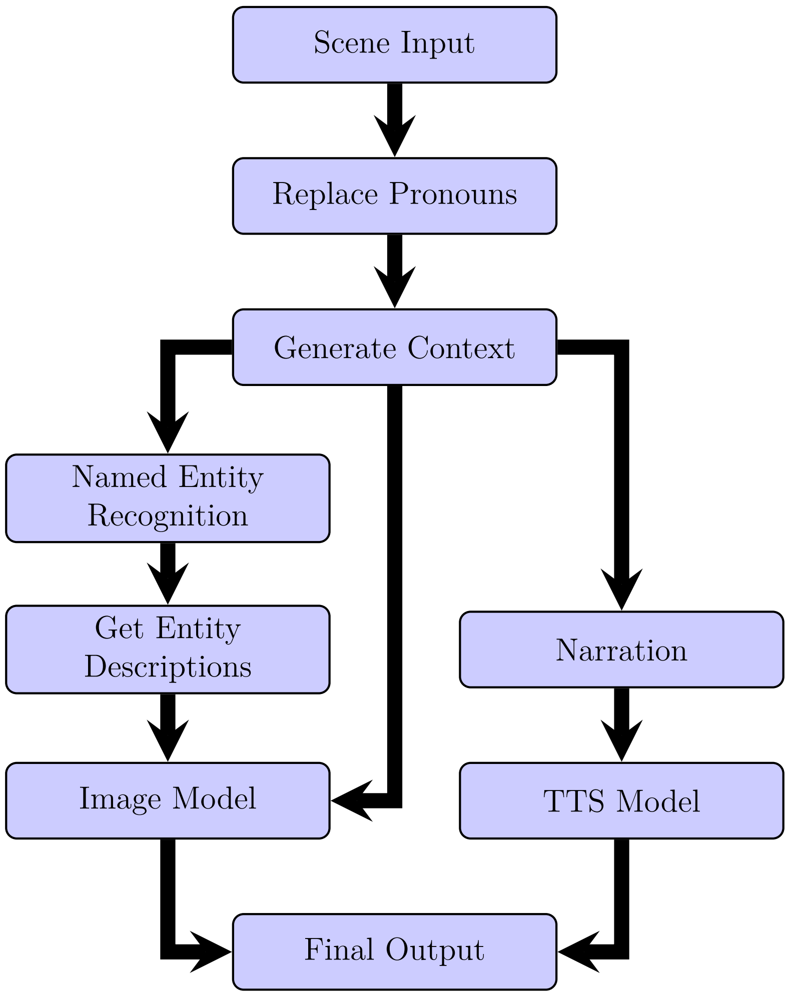

Figure 2: Architecture for Generating Narrated Scenes from Text Using LLMs. This diagram outlines the step-by-step process from initial scene input to the final output involving both visual and auditory modalities. The workflow incorporates stages such as pronoun replacement, context generation, named entity recognition, entity description, and multi-modal output generation using image and text-to-speech models

Your task is to carefully assess whether a new sentence presents a definitive change in the scene of a story, based on the context comprised of the preceding sentences (Prior) and the sentence being analyzed (User). Concentrate exclusively on substantial evidence within the first sentence that unmistakably indicates a transition to a new scene. Your response should be a binary choice, 0 or 1, where 0 signifies no scene change and 1 indicates a definitive scene change (Must return 0 or 1). Given the importance of narrative integrity and flow, always err on the side of marking 0 (no scene change) unless the sentence unequivocally introduces a new environment, time period, or a pivotal shift in the storyline that cannot be interpreted in any other way. Furthermore, to prevent an overestimation of scene changes, only consider marking a scene change if the new sentence introduces elements that are absolutely incompatible with the ongoing scene, such as a different location that requires travel or a time skip that is explicitly mentioned.
Prior:
User:
Response:

Figure 3: Task description for assessing scene changes in a story. The task involves analyzing a new sentence against preceding sentences to determine if there is a definitive scene change. Responses are binary: 0 for no scene change and 1 for a definitive scene change.

applications. From the generated scenes, it is now possible to create the necessary information for all downstream tasks in the chain.

Scene Context

Understanding the scene's core narrative is a crucial step, particularly focusing on what characters are present, doing, and look like. The initial task involves replacing all pronouns in the text with the actual names of the entities they refer to. This is done to help subsequent language models accurately identify who is performing each action. Pronouns, while helpful in reducing repetitiveness in prose, can lead to ambiguities that challenge language models during NER. This ambiguity often makes it difficult for models to correctly identify and attribute actions to the right characters. To address this issue, every pronoun is replaced with the corresponding character's name, which simplifies the text stream for the language models, thereby refining the input for more accurate processing. This method enhances the precision of character tracking and interaction analysis within each scene, setting a more detailed context for subsequent tasks. The given prompt to the model is given in figure 4.

> This task involves meticulously replacing pronouns in a text with the specific named entities they refer to, ensuring clarity and accuracy in reference. The process will utilize a 'prior' text, serving as context to inform the correct identification and substitution of pronouns in the 'input' text section. Ensure that all entities maintain the same name through each and every scene. The goal is to achieve precise and unambiguous communication by explicitly naming the individuals, locations, or objects that pronouns such as 'he,' 'she,' 'it,' 'they,' etc., represent.
> Prior:
> User:
> Response:

Figure 4: Task description for assessing scene changes in a story. The task involves analyzing a new sentence against preceding sentences to determine if there is a definitive scene change. Responses are binary: 0 for no scene change and 1 for a definitive scene change.

The refined scenes are now ready for the next stage of context generation. Up to this point, the process has utilized Chain of Thought (CoT) and resampling to create initial outputs. And now, the mathematics written with Resampling Refinement will be introduced combining both resampling and prompt engineering. To enhance the depth required by image and audio models, an adaptation to CoT is introduced: Expanded Prune Chain of Thought (EPCT). This technique involves resampling to generate diverse outcomes from a single prompt and subsequently condensing these into a singular, optimal response. EPCT enables a LLM to produce detailed outputs or select the most effective result for downstream tasks. While this approach is promising, its effectiveness in this research context remains to be thoroughly assessed, ensuring no detail is overlooked during rapid transitions between sections.

There are many possible implementation of EPCT, and the one utilized within this research is where samples are produced off of the same initial prompt, and a two chain will generalize or summarize all the the given samples back down to a distilled

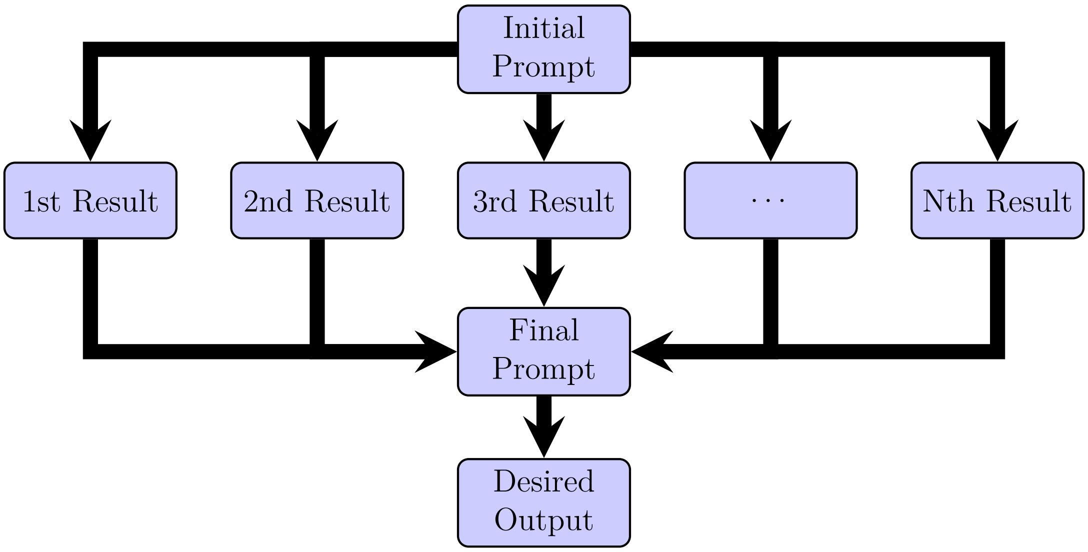

Figure 5: Expanded Prune Chain of Thought (EPCT) technique, which refines diverse outcomes from a single prompt into an optimal response, streamlining model outputs for critical downstream tasks.

version. The structure of this approach can be seen in the following diagram.

And to leverage this strategy of CoT, new prompts should be introduced and given to the LLMs to provide context of the given tasks. These two prompts should initially give the desired output of the model, and then resampling will be deployed to generate the necessary duplicate values for the reduction step in the process. The final prompt should inform the LLM that it will be either enhancing the detail or producing a more distilled version of the result. This new prompt approach will allow the ability to generate enough context for downstream models in this research.

And so once the pronouns are replaced from the scenes in the story, the system prompt to summarize each scene, as well as the scene corpus, are injected to the language model. The prompt should be clear in nature and provide a single solution to the language model to handle. And since the context of each scene will be created in detail and no lost information should occur, the next prompt was utilized to handle the given task. And the prompt shown in figure 6 is the initial prompt in the EPCT method.

Once the results are resampled to the desired amount, the next prompt, shown in

Given the background context provided by previous texts, this task focuses on generating a response that zeroes in on the new information introduced in the current prompt. It's crucial that this response not only acknowledges the foundational context but also significantly elaborates on the fresh details. Aim to weave a response that is intricately connected to the current prompt's specifics, enriching it with a level of detail that is both profound and comprehensive. Ensure that every new piece of information introduced is explored and expanded upon with the utmost depth, embedding it seamlessly within the broader context established by the preceding texts. The response should act as a magnifying glass, highlighting and elaborating on new details in a way that is relevant and directly tied to the query at hand. While the background context must inform the response, the focus should remain on the novel elements, exploring their implications, nuances, and potential impact with precision and detail. Maximize the use of available tokens to paint a vivid picture that is rich in detail and full of insight, ensuring the response is not only relevant but also adds substantial value and depth to the discussion. The goal is to create a response that, within the limit of 4048 tokens, is as informative and detailed as possible, providing a thorough exploration of the new information while situating it within the larger narrative provided by the previous texts.
Prior:
User:
Response:

Figure 6: Initial step in the EPCT method, focusing on generating detailed responses that both recognize foundational context and expand on new information for each scene.

figure 7, will take in all of these and then return a final result that will encapsulate the entire meaning behind each scene. The prompt explains the context the LLM is working in and the goal of returning a detailed description of the scenes given.

And the input in this example will be the concatenated responses of the previous step in the EPCT approach. The results are long and detail account of each scene, which can be leveraged to create the list of known entities, the description of each entity, the images to illustrate the scene, and finally the text for narration.

This task requires the generation of extraordinarily detailed and intentionally redundant texts to ensure an image model can produce consistent visuals across various prompts. The goal is to saturate the description with vivid, precise, and repeated visual details that paint a clear and unambiguous picture of the scene or subject. This means going beyond mere description to include nuances of light, texture, color, and spatial relationships. Think about how each element interacts with the others, and how they collectively contribute to the atmosphere of the scene. Use metaphors, similes, and descriptive adjectives liberally to add depth and dimension to your descriptions. Remember, the aim is not just to describe but to evoke a sensory experience that is almost tangible. When discussing objects, characters, or landscapes, consider their appearance, emotion, and significance within the scene. Where relevant, include background information that might influence the image's interpretation but weave this information seamlessly into the visual description to maintain focus. The narrative should be so detailed that someone could draw the scene with accuracy based solely on your text, capturing not only the physical attributes but also the mood and emotions conveyed. Your text should serve as a comprehensive guide for the image model, allowing it to generate images that are remarkably consistent, regardless of how many times it is prompted. Redundancy is key; do not shy away from repeating important visual elements to ensure they are captured in the generated images. Be as detailed as feasibly possible, aiming to create a text-rich canvas from which the image model can draw inspiration.
User:input
Response:

Figure 7: Task description for generating detailed and redundant texts to ensure consistent visuals from an image model. The goal is to create vivid and precise descriptions that capture nuances of light, texture, color, and spatial relationships, evoking a sensory experience. This comprehensive narrative guides the image model to generate consistent images across various prompts

Narration

Narration is a vital component in storytelling, serving as the backbone that supports the entire structure of the narrative. It informs the reader's understanding, guiding their emotions and perspectives throughout the tale. The way an author chooses to narrate a story can profoundly influence its tone and the overall reading experience.

Effective narration imbues the story with depth, making it compelling and thought-provoking. Thus, creating a precise and evocative narrative prompt is essential to ensure that the story resonates well and maintains its intrigue. This prompt below shall give enough information to provide that level of importance.

> Using the information provided in the previous texts as background context, generate a response that focuses on the new information presented in the current prompt. Please ensure the response is relevant to the current prompt and avoids repeating details from the earlier texts. Aim to weave a rich narrative that explores the implications and nuances of the new information, enhancing the story's depth and emotional resonance. Previous Texts for Reference (do not repeat these details):
> Note: The response should build upon the context provided by the previous texts but should be specifically about the new elements introduced in the current prompt. The narrative should be insightful and detailed, providing a thorough exploration of the new information within a limit of three sentences, without being repetitive.
> User:
> Response:

Figure 8: Directs the model to generate an in-depth narrative based on new information, encouraging detailed exploration without repeating background context.

Character Description

And to branch away from the narration, the focus will be on the illustration of the story. The next important step in this pipeline is the character descriptions. Thus, it is required to find the named characters present in each scene. And Named Entity Recognition will be instrumental in the story stream to the characters for each scene. This step will also employ resampling from the LLM, and this situation is more important since the response is not a singular token, but multiple with conditions on the previous tokens provided.

The prompt that shall return the necessary list of entities that are present in this scene. This prompt aims to give context to the LLM that it will in turn provide the name of the present characters

This process will generate a list of named entities after analyzing multiple samples from the given prompt. To mitigate the likelihood of hallucinated entities by the

Analyze the given text to identify and extract all character names mentioned. Create a Python list named 'characters' containing each character's name as a separate string. Ensure to include every character mentioned by name, regardless of the frequency of their appearance. Exclude entities not explicitly named. In the case of no character names being present, return an empty list. Prior user input should be incorporated to update the list with any character names introduced earlier in the story. Note: Only provide the Python list.
Previous Character List:
User Input:
Response:

Figure 9: Analyzes text to compile a list of character names using Python, updating an existing list based on prior input and incorporating resampling techniques to ensure comprehensive and accurate extraction

Language Model, a threshold, denoted as p, will be applied. This threshold serves to filter and retain only the most probable entities from the resampled data. Consequently, the final list will encompass the predicted entities within each scene, setting the stage for the subsequent task of acquiring detailed descriptions for each identified entity.

With the named entities established for each scene, we can now progress to deriving detailed character descriptions, which are pivotal for the final segment of the project—scene illustration. To ensure that the visual attributes and characteristics of each character are accurately captured, the model will focus on the first n appearances of characters in each scene. The Expanded Chain of Thought Reduction (EPCT) technique is employed again to obtain these detailed descriptions, which are essential for ensuring consistent visual representations across the illustrations. The initial prompt used for this task is illustrated in the following figure:

the prompt in figure 10 is designed to intake a specified input alongside the name of the entity to be described. This approach is particularly effective in scenarios where scenes are populated with multiple characters, prompting the model to focus solely on providing a description for the targeted entity. Subsequently, the next prompt will "prune" the samples and select the most relevant character descriptions into a

This task involves creating detailed, repetitive texts that accurately describe the physical traits of {entity}, including features, stature, attire, and distinguishing marks. Use clear, descriptive language to ensure the image model can consistently and accurately depict {entity}'s appearance. The description should serve as a comprehensive guide, enabling the creation of detailed visuals that faithfully represent {entity}.
Input:
AI:

Figure 10: Creates detailed, repetitive descriptions of an entity's physical traits, which will guide the image model in producing consistent and accurate visual representations. This detailed character profiling is critical for ensuring that the visuals generated by the model faithfully reflect the described features of the entity.

concise, yet comprehensive character profile.

Take the input and summarize it into one extremely detailed and short description of entity and only that entity. Do not provide anything else besides the description.
Input:
Response:

Figure 11: Involves condensing extensive character descriptions into a singular, precise depiction of an entity, focusing solely on essential traits to guide accurate visual representation.

With these enriched character descriptions at hand, along with the detailed contexts of the scenes in which these characters appear, we are well-prepared to execute the final phase of the project: the illustration of the narrative. For this task, a specialized prompt will collate only the relevant character descriptions for each scene, coupled with the scene descriptions. This streamlined prompt, simpler than its predecessors, is crafted to deliver just the right amount of detail to the image model, ensuring clarity and precision in the visual rendering of the story.

This approach concatenates the character descriptions as well as the scene context to prompt the image model to create the desired results. So in the chain, there is a check to evaluate which characters are present in which scenes. The tracking of characters as well as knowledge of the current scene ensures a clear transition from

```
scene:{scene_context}
character descriptions:{input}
```

Figure 12: This prompt combines the scene context and character descriptions to guide the image model in generating accurate visual representations. It ensures the scene's coherence and character details are well-integrated.

textual analysis to visual representation, crucial for the successful adaptation of the narrative into illustrated format.

FINDINGS

The implementation and test of the methodologies outlined in the previous sections, exploring their efficacy and reliability in achieving desirable outcomes for each step. This segment of the research focuses on an extensive array of tasks derived from the original corpus of text, including sentence delineation, scene context generation, narration, and image generation. Each task is crucial for reconstructing the narrative in a format suitable for analysis and visualization, employing advanced generative models. The process begins by dissecting the corpus into sentences, which are then methodically analyzed to establish their scene grouping. These delineations serve as a foundation for generating coherent narrative scenes, which are subsequently transformed into narrated segments and visual representations. This systematic approach ensures that the transition from textual content to multimedia output retains the integrity and continuity of the original narrative, addressing both the challenges and potentials of leveraging large language models in literary analysis and content creation.

To execute the methodologies from the previous sections, the pre-trained LLM leveraged is OpenAI's ChatGPT 3.5-turbo. For TTS and image models, this research utilizes OpenAI's TTS and DALL-E models, respectively, for their specific tasks.

Scene Delineation

During the scene delineation step of this research, the input will be given to the model in addition to any possible prior sentences in the sequence to ensure the language model is able to decide if and when new scenes are introduced. An example of this can be seen in the figure 13 below.

Each of these prompts will either return 0, 1, or 'neither'. Results classified as 'neither' will be discarded and then resampled to ensure a robust dataset. This

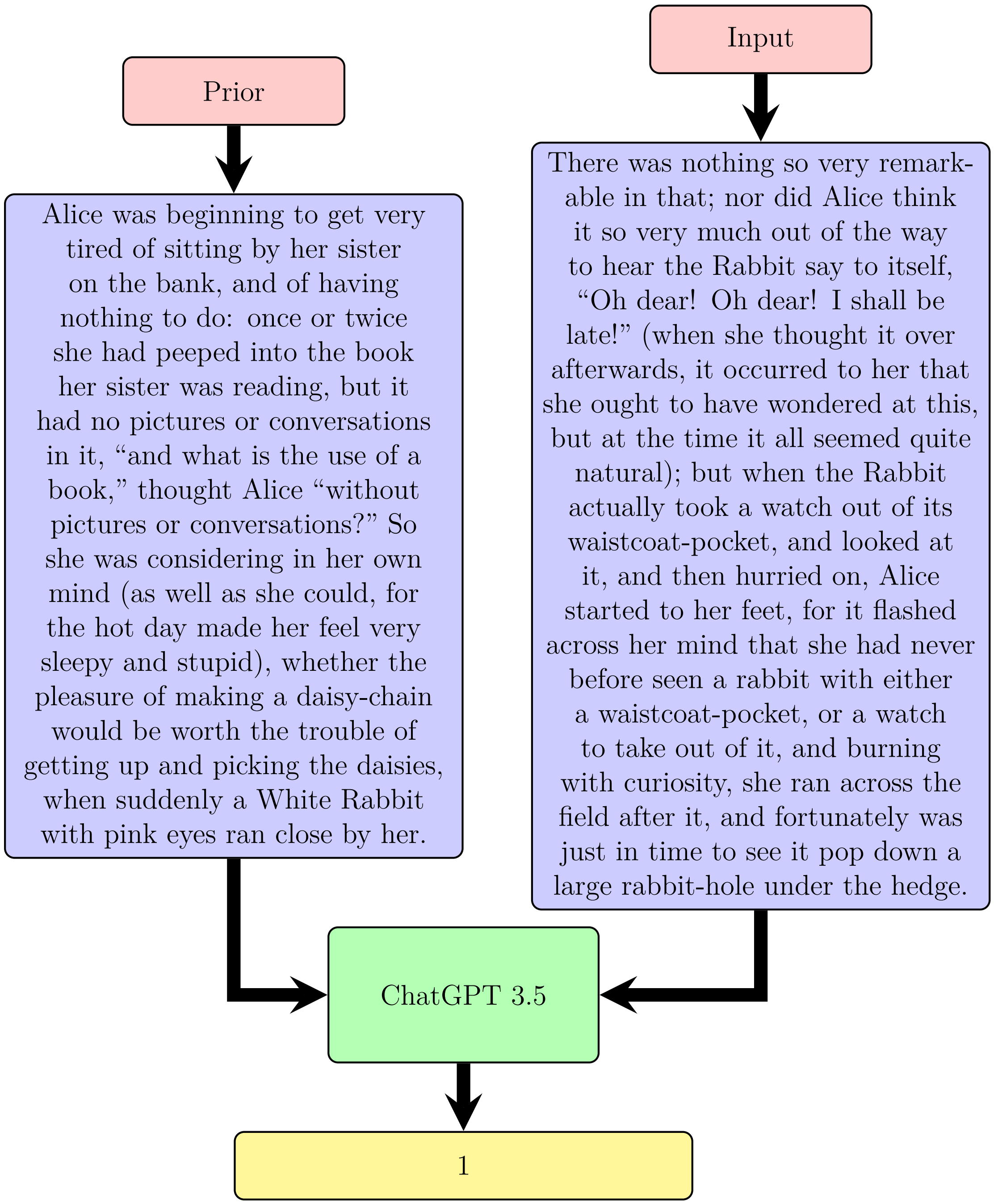

Figure 13: The processing of textual input using ChatGPT 3.5. The model utilizes prior context from previous sentences as well as the current sentence to predict if a scene change had occurred.

resampling is conducted for each sample to accurately predict the occurrence of scene changes within each sentence of the story. A sample size of 5 was implemented, and a cutoff threshold of 80% was utilized to minimize the number of scene changes, aiding in more effective context generation for downstream tasks.

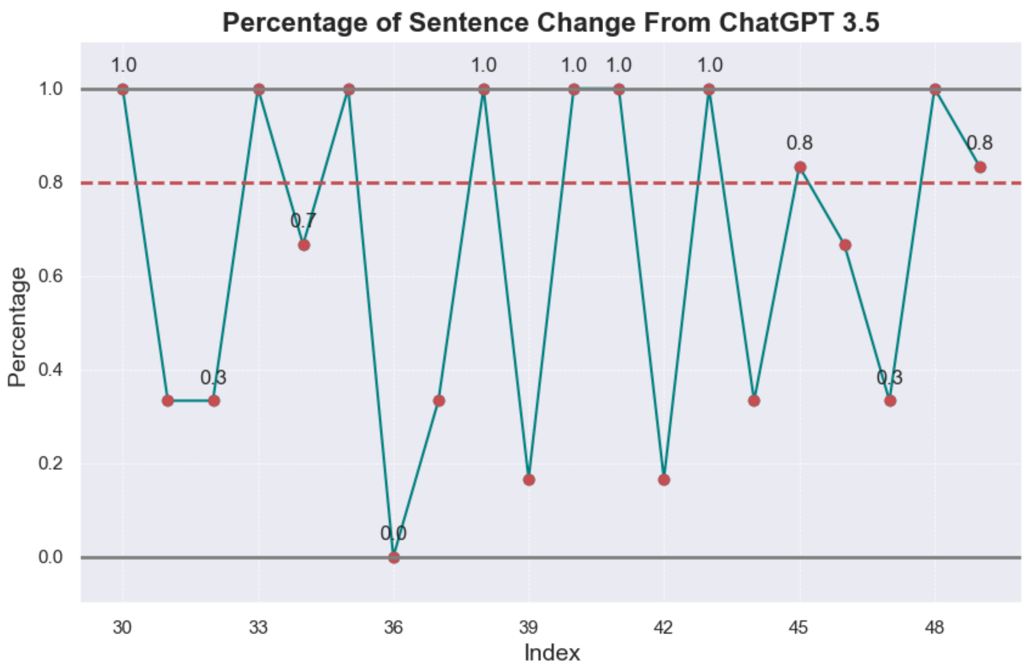

Figure 14: Graph depicting the sampled averages of sentence changes determined by ChatGPT 3.5-turbo, indexed by sentence position. Each point represents the percentage of identified scene changes, with a red dashed line indicating the 80% threshold

And figure 14 shows that the prediction is likely to return a scene change for some of the scenes, and we should increase our thresholds to ensure each scene has enough information in it to reduce hallucinations and poor contexts for narration and image generation. With a increased sample size, and a smart selection of the threshold, it is clear to see that the resampling method for LLMs gives robust analysis into results of an LLM.

The exploration of Bootstrapping reveals its effectiveness in analyzing the distribution of output means from the LLM. With the outcomes to generate a series of bootstrap samples, each of which is used to compute estimations of the mean. These estimations are then employed to create a distribution of these estimators, providing insights into the variability and reliability of the model's performance. By analyzing thousands of bootstrap samples, the mean of the model's output distribution can be estimated with greater confidence.

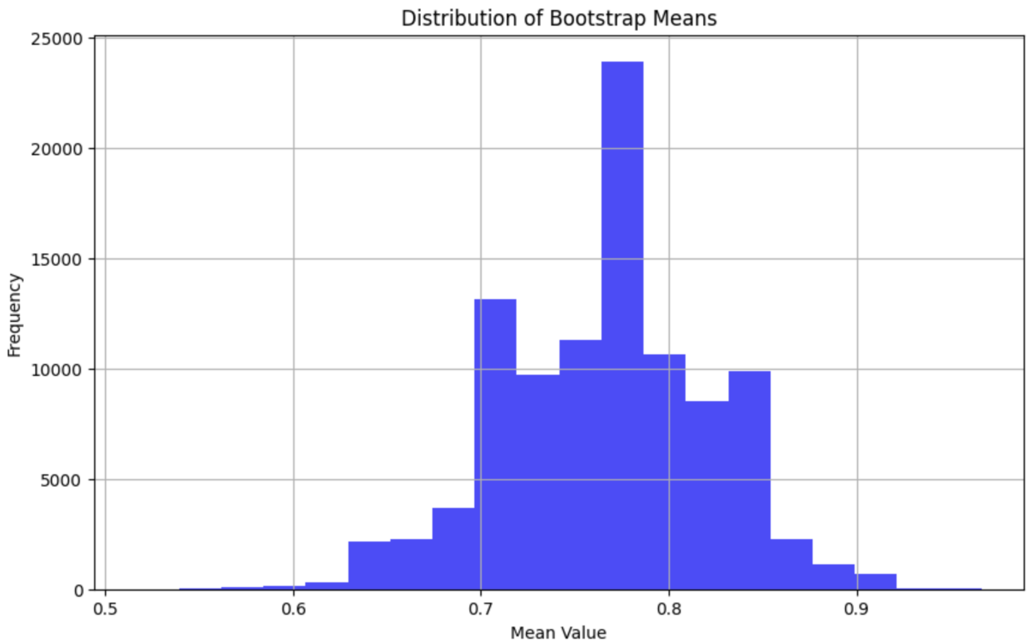

Figure 15: Distribution of bootstrap means with adjusted bins. This histogram illustrates the variability and concentration of bootstrap sample means.

This plot visualizes the distribution of the bootstrap means, focusing on the analysis of the transformer model's performance with binary outputs. The histogram showcases the concentrated distribution of mean values around 0.8, indicating a strong tendency of the model to consistently predict a specific class in binary scenarios. The distribution's shape, with its peak at 0.8 and tails extending towards lower and higher means, provides a visual representation of the variability in the model's output.

Context Generation

These scenes are then grouped with the previous scene if the current prediction is not true. The scenes will have their pronouns replaced with the actual named of the present entities and run through EPCT to provide detailed scene context. The structure for the pronoun replacement is quick direct and the model is fed in the input, which is the current scene, and the system prompt as shown in figure 4, tasks the model to remove all pronouns.

The model shown in figure 16 handle the replacement of pronouns effectively, and after this is done through each scene iteratively, the detailed contexts generated are then processed through the EPCT implementation. This process produces multiple samples of the same prompt, and the pruning step compresses everything into a more detailed response for downstream systems.

Once the scenes are fully processed and the context is well-defined, the next step is to ensure that these scenes are coherent and accurately reflect the narrative's flow. The refined scene contexts are crucial for subsequent tasks and an example of these are shown in figure 20, such as narration and visual representation, ensuring that the generated outputs are rich in detail and maintain the integrity of the original narrative. This method enhances the depth required by image and audio models, setting a strong foundation for creating comprehensive and coherent multimedia outputs.

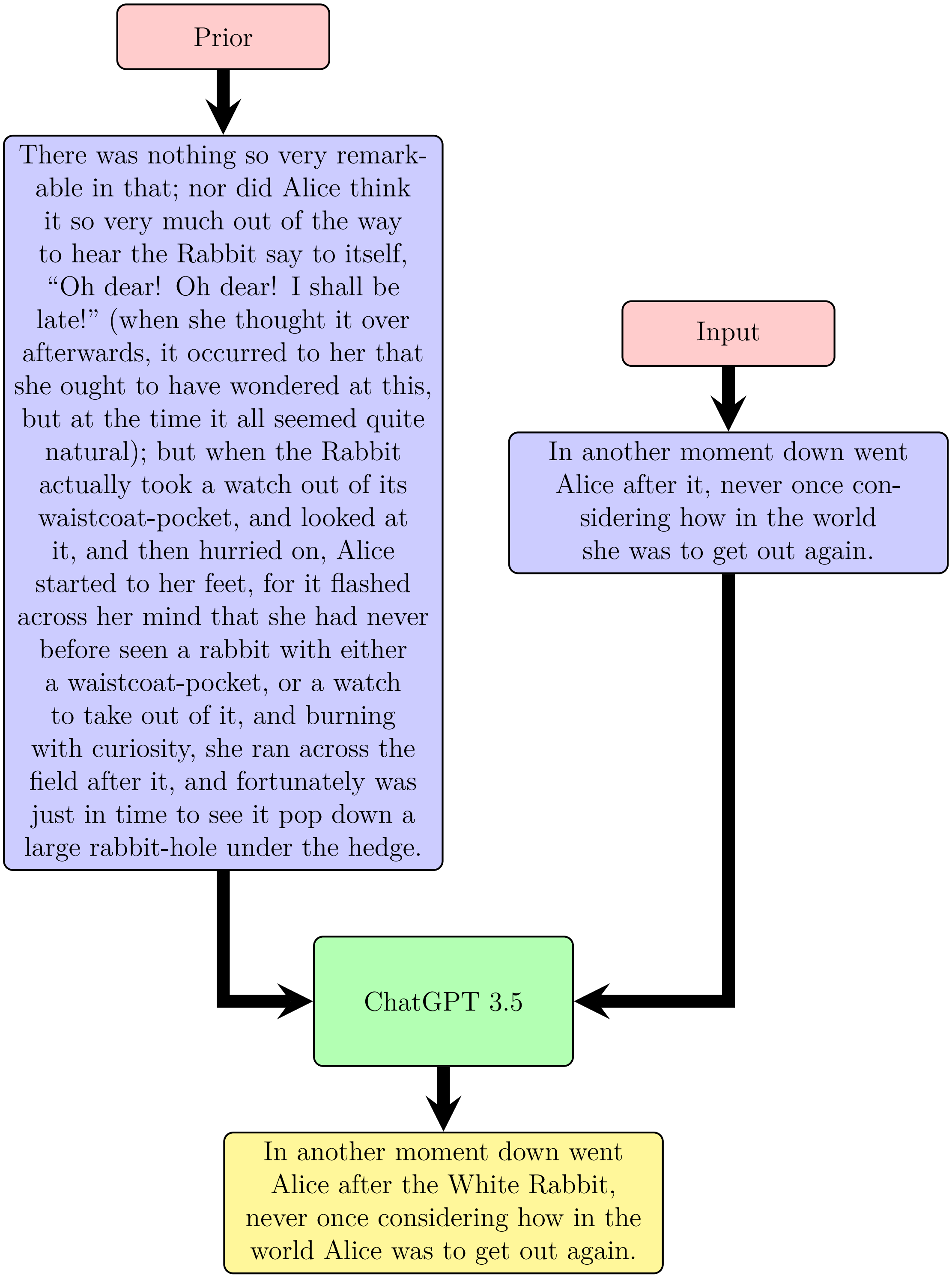

Figure 16: The processing of textual input using ChatGPT 3.5. The input is fed into the model with the system prompt to remove any pronouns.

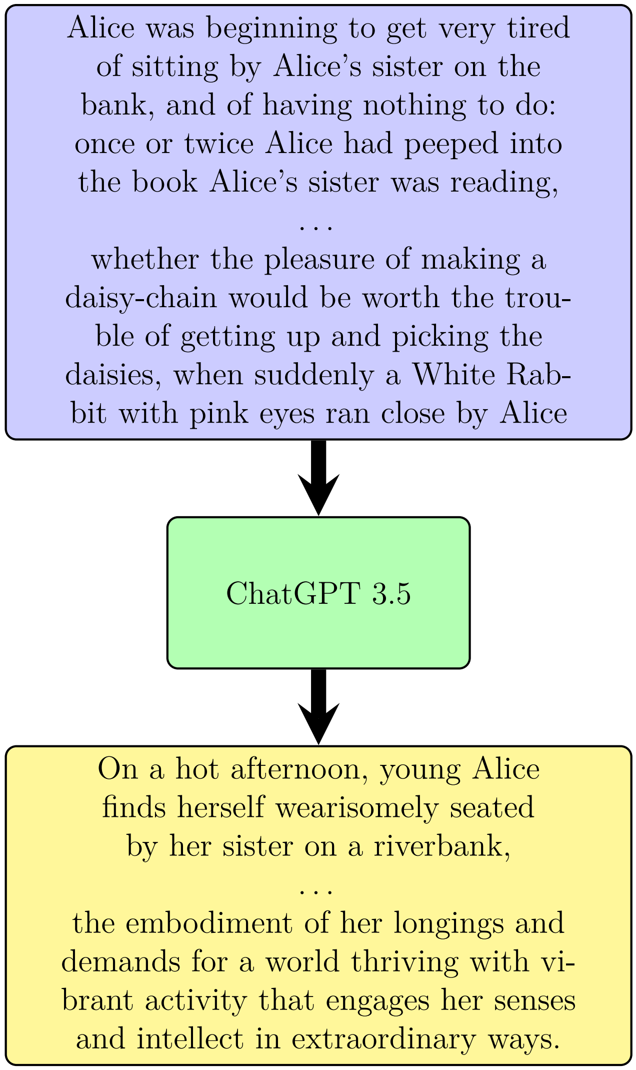

Figure 20: Implements EPCT for creating detailed scene descriptions. Taking in one scene expanding it within EPCT and then back into one result.

Narration

Once the scene context is refined, it can be used to generate the text for narration. This text is then input into the TTS model developed by OpenAI. The process starts with the enhanced scene context, incorporates narration generation, and finally, feeds this narrative text into the TTS model. This chain effectively transforms the refined text into the spoken word, mirroring the voice and intonations of the story as intended in the book.

The effectiveness of this chain in processing audio from the narrative text was

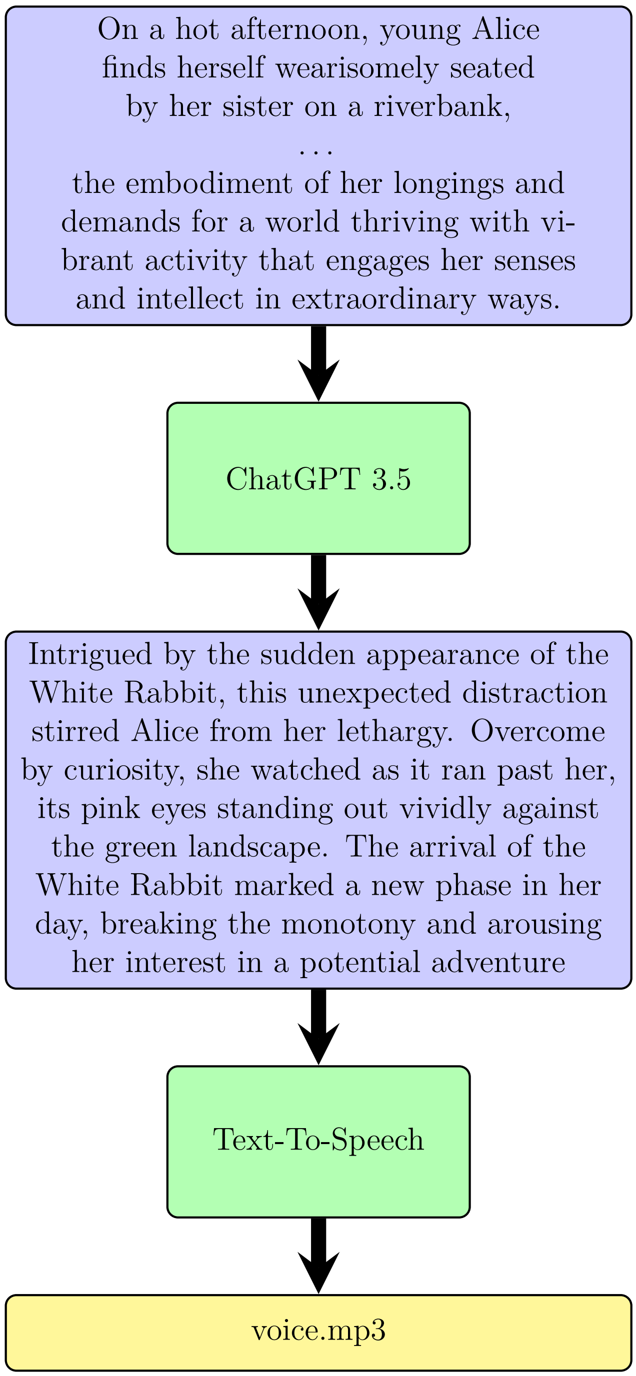

Figure 18: Taking in the scene context, to convert to the text narration, and finally to a mp3 of the voice talking.

notable. The transition from text to speech maintained a high fidelity to the original tone and emotion of the story, demonstrating the TTS model's capability to produce clear, expressive audio that faithfully represents the narrative content. This leave the chain to now work on handling character descriptions for image generation, and then finally the scene generation.

Character Description

To obtain character information in this information, the iteration through each scene and leveraging LLMs once again to produce the NER for each character present in the scenes. Resampling will once again be employed to increase the efficiency of traditional uses of LLMs for entity recognition. The follow figure 19 demonstrates the difference between the classification results of LLMs to resampling expecting a specific result. If the model generates a non-conforming result, that response will be discarded and another samples will be produced.

As it can be seen in 19, the response is not a single classification, but a list of present entities. This approach allows for the handling of diverse narrative elements within the text, ensuring that each character's presence or absence contributes significantly to the richness of the narrative context.

The data shown in figure 21 highlights the presence percentages for four characters: the White Rabbit, Dinah, Alice, and Alice's sister. The graph clearly delineates the characters' involvement, with the White Rabbit and Alice showing consistent, high-frequency appearances across the samples. In contrast, Dinah and Alice's sister appear less frequently, suggesting their peripheral roles in the storyline.

This binary result of presence-absence, where characters are marked as either completely present or absent in each sample demonstrates the LLM is extremely confident if the present characters of the story. This technique, derived from the resampling methods discussed in the Methodology section, ensures a precise characterization of

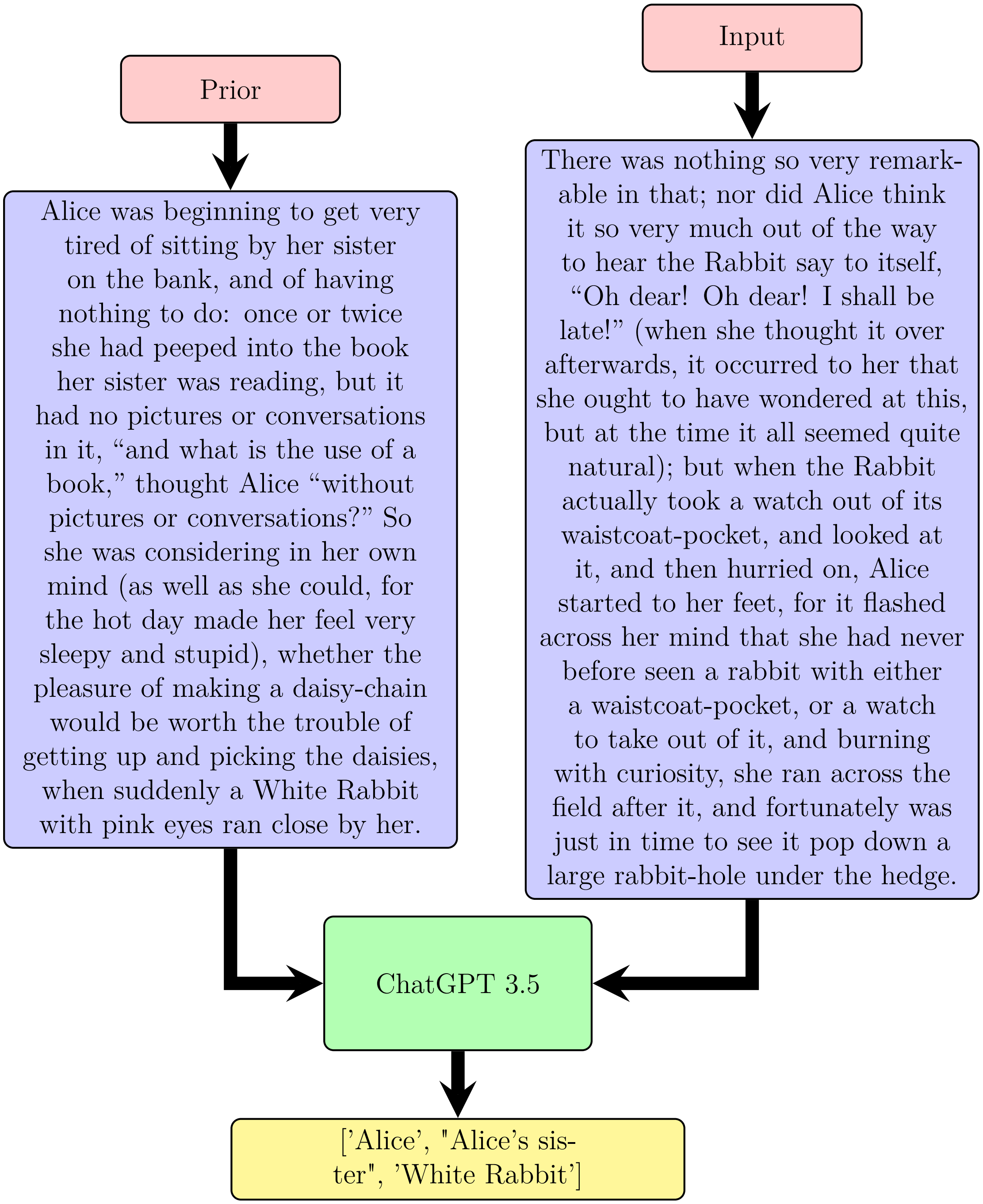

Figure 19: The processing of textual input using ChatGPT 3.5. The model utilizes prior context from previous sentences to predict the present entities in a given scene.

narrative elements. This not only supports the generation of contextually rich narrative outputs but also enhances the predictive capabilities of automated systems in content creation.

Now, with the list of entities in the story and when they appear, it is possible to create the descriptions of each of the characters leveraging EPCT.

Image Generation

Since the image generations will take in each scene context as well as the character description, a trade off appears when attempting to give the image model, OpenAI's Dall-E enough information on the scene and a character description. This results in inconsistent character visualization or lost of the context of the scene at hand. The character comparison in figure 22 have been generated and are present in different scenes, and as it can be seen the illustrations of Alice do not maintain any structure or style guides. The prompts have been experimented with to see if this could remedy this issue, but these attempts returned similar results.

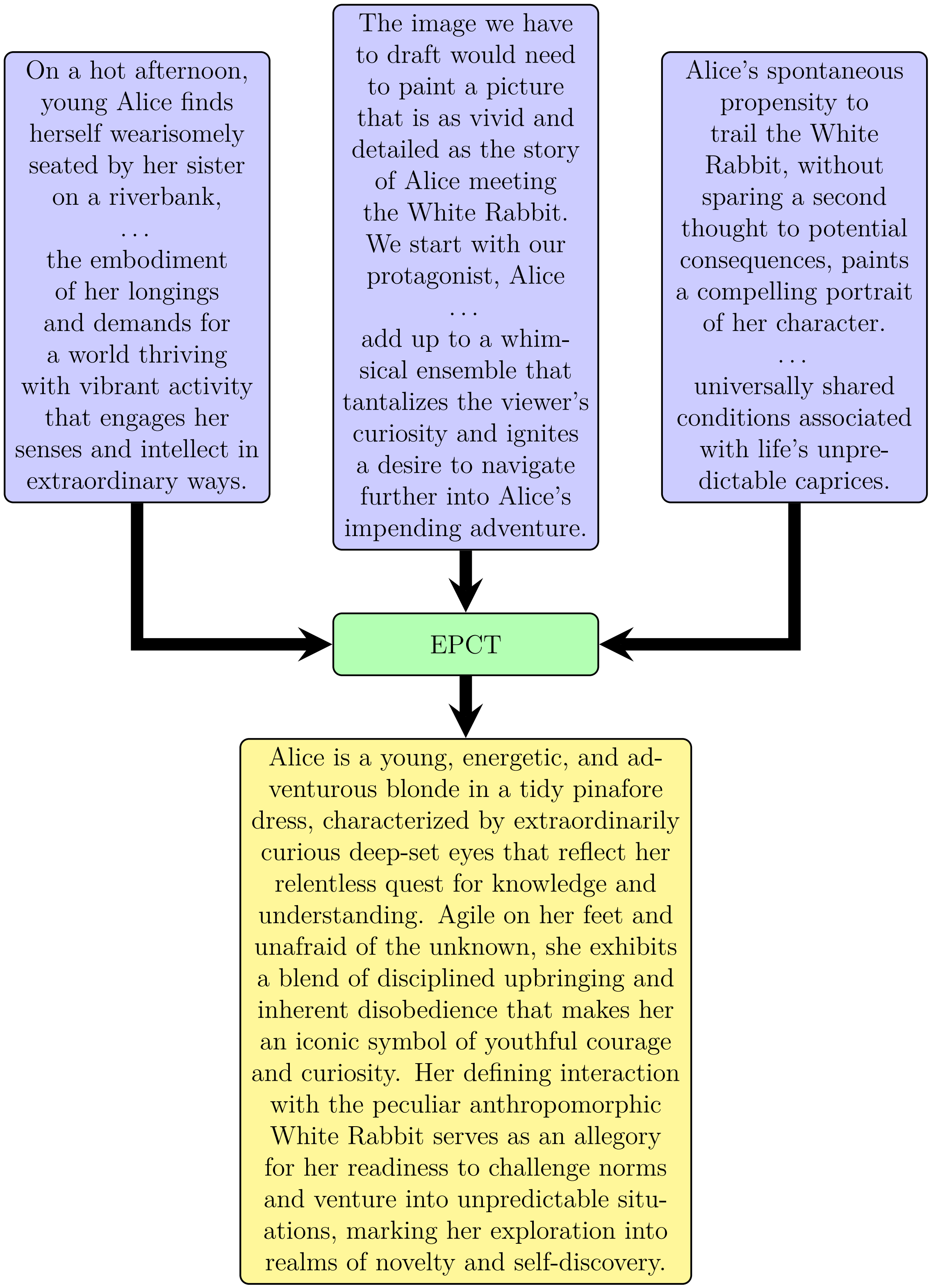

Figure 20: Integrating and refining inputs into a singular, comprehensive narrative

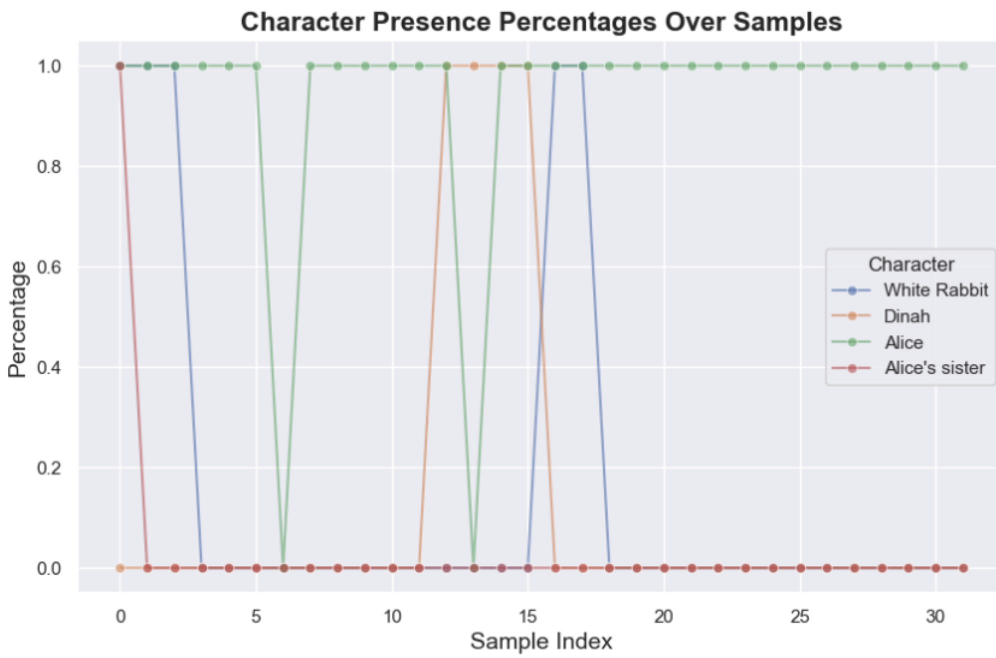

Figure 21: These results showcase the models ability to validate when certain entities are present in each scene. It can be see that Alice is present in almost all scenes, and all other characters are only present partly

Context is extremely import to image models and Dall-E's model is a stable diffusion model, and so since the input prompt includes both the sense and character description, the model loses context in the description of the character as well as the limitation of diffusion models where even with the same prompt results can inconsistent. This limitation likely lies within the architecture of the diffusion model where the initial state is random and slowly creating an image that using the prompt to guide the image creation(Sohl-Dickstein, Weiss, Maheswaranathan, & Ganguli, 2015).

Language models' ability to transform textual narratives into structured, analyzable formats and multimedia content is possible. The Text to Speech model successfully delivers desirable results. However, this cannot be said for the Diffusion model. Dall-E's problem lies in the architecture of the model, and this issue only compounds when the model cannot achieve consistent results between scenes. Addressing these architectural challenges and enhancing model robustness will be crucial for future applications, aiming to improve reliability and expand the practical uses of generative

Figure 22: Illustrates the inconsistency in visual outputs generated by a DALL-E, depicting four versions of Alice from "Alice in Wonderland." Each rendition varies significantly in style and detail

models in creative industries.

CONCLUSION

In this thesis, we explored the capability of foundation models, specifically focusing on their application without fine-tuning, to transform textual narratives into structured, analyzable formats and multimedia content. Through our extensive experimentation with resampling techniques, we demonstrated that these models could effectively parse and reconstruct narratives into detailed scenes and character descriptions, facilitating the generation of rich multimedia content.

However, challenges were evident, particularly in the domain of image generation using Dall-E. The model's architecture, while robust, showed limitations in achieving consistent visual outputs across different prompts. This inconsistency highlights a critical area for improvement within the underlying mechanisms of diffusion models. Despite their robustness, diffusion models such as Dall-E currently struggle with consistency across different prompts, which remains a significant limitation. Addressing these consistency issues with improved implementations could enhance their reliability and effectiveness.

Future research should focus on integrating enhanced model architectures and considering minimal fine-tuning to address discrepancies. These advancements could be tailored for creative industries, potentially enhancing utility and improving multimedia content reliability. This approach aims to refine performance and extend the applicational scope, leading to richer, more precise multimedia narratives.

This thesis addresses the challenges of converting written narratives into films by showcasing the potential of Generative AI to enhance adaptation fidelity and efficiency. Automating and refining scene segmentation and narration through LLMs promotes a more consistent and accurate conversion of text into visual and audio formats. Combining traditional techniques with cutting-edge methods enriches the adaptation process and could shift the roles of creatives in the industry, focusing more on artistic expression. The exploration of advanced NLP and GenAI technologies opens new prospects for film adaptations, capturing the essence of original narratives more effectively.

LIST OF REFERENCES

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., & Wang, G. (2023). GPT-NER: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428v4*. Retrieved from https://arxiv.org/abs/2304.10428v4

Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., & Wang, G. (2023). Text classification via large language models. *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.emnlp-main.22

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. Retrieved from https://arxiv.org/abs/1706.03762

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2023). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*. Retrieved from https://arxiv.org/pdf/2402.07927

Tan, Q., Qin, L., Soong, F., & Liu, X. (2021). A Survey on Neural Speech Synthesis. *ACM Transactions on Audio, Speech, and Language Processing*, 29(1), 1-35.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. Retrieved from https://arxiv.org/pdf/1503.03585

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1), 1-26. Retrieved from https://doi.org/10.1214/aos/1176344552

Shepp, L. A., & Olkin, I. (1978). Entropy of the sum of independent Bernoulli random variables and of the multinomial distribution. SIAM Journal on Applied Mathematics, 34(4), 676-682. Retrieved from https://epubs.siam.org/doi/10.1137/1123020