Story GenAI: A Comprehensive AI-Based Framework for Personalized Kids Story Generation Using Natural Language Processing and Deep Learning

Mr. Fahad Ahmed Salmani
Dept. of Artificial Intelligence & Data Science
University of Mumbai, Mumbai, India

Mr. Pravin Santosh Tiwari
Dept. of Artificial Intelligence & Data Science
University of Mumbai, Mumbai, India

Mr. Dhiraj Pal
Dept. of Artificial Intelligence & Data Science
University of Mumbai, Mumbai, India

Mr. Gaurav Shaw
Dept. of Artificial Intelligence & Data Science
University of Mumbai, Mumbai, India

Asst. Prof Oyendrila Samanta
Dept. of Artificial Intelligence & Data Science
University of Mumbai, Mumbai, India

Abstract: Storytelling is a vital tool in the formation of cognitive capabilities, imagination, and language skills among children. The current methods of storytelling are based on content that is fixed and narrated manually by a human being, thus failing to scale up and customize the experience. This research proposes StoryGenAI, a holistic Artificial Intelligence-powered framework that provides personalized children's storytelling services through real-time generation, based on advanced Natural Language Processing (NLP) and deep learning models. StoryGenAI uses transformer-based neural networks, such as Generative Pre-trained Transformers (GPT), to develop well-written, coherent, context-dependent, and interesting stories. The framework employs prompt generation based on input from the user, semantic control strategies, and post-processing techniques to facilitate age-appropriate storytelling content generation. Moreover, StoryGenAI integrates speech synthesis to provide interactive storytelling services.The results of experiments prove that the model is highly coherent (88%), grammatically accurate (90%), and highly rated (87%) among its users. The contributions of the proposed framework include enhancing intelligent educational tools through the use of AI.

Index Terms—Artificial Intelligence, Natural Language Processing, Story Creation, Deep Learning, Transformers, Educational Applications, GPT.

I. Introduction

Since ages, storytelling has formed an essential part of human society and plays a vital role in the early stages of life. Storytelling helps develop creativity, linguistic skills, and emotional intelligence. Traditional storytelling is confined to the limitations of human input and effort.

In the current era of technology, where children have more interactions with computerized applications, it is observed that although various applications are available which offer stories to children, yet they are mostly pre-defined in nature without customization as per the user's choice regarding characters, theme, or moral lesson.

The advancements in Artificial Intelligence have brought about some significant developments in the area of automatic content creation. Deep learning algorithms are capable of analyzing patterns within extensive text databases and generating meaningful stories.

With the advancement in transformers, GPT is now capable of generating high-quality content on the basis of user-defined prompts. The proposed system seeks to leverage the mentioned technologies to create a storytelling system that can generate personalized stories in real-time. The user is able to set parameters including the types of characters, themes, morals of the story among others.

Furthermore, the system uses contextual awareness and semantics for maintaining coherent logic within the story plot. Language models are used for analyzing connections between various components of the story and ensuring the story flow remains logical and consistent. As a result, the stories created using the system closely match those written by humans.

Also, the system features optional multimodal capabilities that involve text-to-speech functionalities. Such feature is useful for enhancing user experience by providing auditory stories. This will be especially useful for younger children who might not enjoy reading and would prefer listening to stories. Additionally, such feature will assist students with difficulties related to reading skills.

All things considered, the described solution called StoryGenAI demonstrates the potential for creating an advanced system that will enable intelligent storytelling among other capabilities. Integrating artificial intelligence with education-related objectives proves useful in achieving the goal

The key contributions of this research include:

- Development of an AI-assisted storytelling framework that can be used for storytelling among children.
- Design of transformer models for automated narrative generation.
- Prompt Engineering to enable controlled storytelling.
- Development of personalization capabilities in the model.
- Multi-modal interactions supported (text and voice).

I. Literature Survey

In the last few decades, automated story generation technology has seen major improvements. The earlier generations were rule-based and relied heavily on template generation methods to develop a grammatically correct story.

Statistical approaches such as n-grams improved the process as they focused more on prediction of the word sequence probability. Though effective, these models suffered from challenges of context maintenance.

Introduction of neural networks was another major milestone. Recurrent Neural Networks (RNN) and LSTM networks were extensively used in generating text. These models had an improvement over the earlier approaches when it came to contextual information handling.

Transformers emerged as another revolutionizing architecture. They rely on the principle of self-attention to process sequences entirely at once.

Generative models like GPT have shown remarkable success in performing text completion, dialogue generation, and storytelling tasks. Such models learn from huge data and create coherent narratives.

In recent years, several studies have tried to integrate AI-driven storytelling systems in educational institutions. However, most of the systems lack real-time adaptation, personalization, and interactive capabilities.

These shortcomings have been overcome using the proposed StoryGenAI system, which incorporates state-of-the-art Natural Language Processing techniques with user-oriented personalization and interactivity features.

The research into multimodal storytelling has become popular too. Contemporary technologies combine text with audio, pictures, and even animated visuals to deliver immersive experiences. Specifically, the process of conversion of the generated text into audio files using the text-to-speech technologies has proven to be an effective practice, providing easy access to storytelling content for younger children who cannot read yet and older people who may have trouble reading.

Nevertheless, there are still issues that require further attention from researchers within the sphere of automatic story generation. The issues include but are not limited to long-term coherence, repetition avoidance, logical consistency, and ethical safety. To address such issues, the process of developing and implementing filtering methods is required.

The proposed approach for automatic story generation in children includes the combination of transformer models for text generation with prompt engineering, content filtering, and customization. The described solution provides an opportunity to move the current level of technology development to the next stage and to create efficient automated kids story generation technologies.

II. System Architecture

The proposed StoryGenAI system uses a four-layered architecture comprising the following layers:.

A. *User Interface Layer*

This layer facilitates intuitive interaction with the user and requires user input in the form of:

- Type of character (animals, fantasy, kid, etc.)
- Theme (adventure, moral, fantasy)
- Length of the story
- Moral/Lesson.

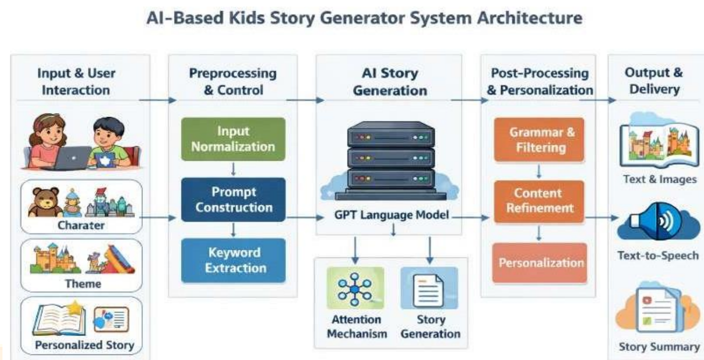


Fig. 1. AI Based Kids Story Generator System Architecture

B. *Preprocessing Layer*

This layer pre-processes user inputs to provide processed data to the AI model.

- Text normalization
- Tokenization
- Prompt construction

C. *AI Generation Layer*

It is the core module of our system.

- Language Model (Transformer/GPT)
- Context representation generation
- Sequence prediction

D. *Delivery Layer*

The delivery layer is responsible for displaying the story via.

- Text interface
- TTS interface
- Download/export capabilities.

III. Methodology

The system uses a systematic pipeline that is made up of several stages. In order to facilitate efficient and quality story creation, the StoryGenAI framework uses a methodical process that involves different stages to create a coherent and relevant output. These stages cover all the processes involved from the beginning to end in terms of generating a story for a user.

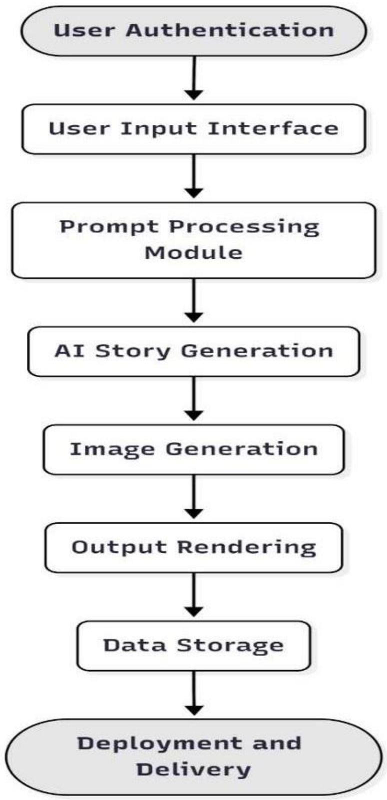


Fig. 2. System Methodology and Execution Flow

The operational pipeline of the proposed storytelling framework follows a strict, sequential architecture designed to transform raw user intent into a fully realized, multi-modal picture book. The system workflow is divided into eight primary computational stages, as illustrated in the methodology flowchart.

1. User Authentication

The system initiates at the security layer, where user identity is established and verified. Utilizing JSON Web Tokens (JWT) alongside bcrypt password hashing protocols, this stage ensures secure session management. This isolates user data, allowing for personalized experiences and secure curation of individual story libraries.

2. User Input Interface

Once authenticated, the user interacts with the frontend Single Page Application (SPA). This layer acts as the primary data acquisition module, capturing highly specific narrative parameters. The interface records demographic and thematic variables—such as the protagonist's name, age, gender, and the desired story genre—which serve as the foundational constraints for the subsequent generation phases.

3. Prompt Processing Module

Raw user inputs cannot be fed directly into the core neural networks without structural refinement. In this middleware stage, the backend API intercepts the acquired parameters and programmatically structures them into a highly optimized, rigid narrative prompt. This module enforces the necessary 5-beat story arc (Setup to Resolution) and calibrates the syntactic complexity constraints based on the provided target age.

4. AI Story Generation

This node represents the core Natural Language Processing (NLP) engine. The processed prompt is transmitted to a foundational Large Language Model (e.g., Llama 3.3 70B), which synthesizes the continuous prose. Immediately following the primary synthesis, a secondary formatting model segments the raw narrative into a structured JSON payload, logically dividing the text into distinct pages while ensuring age-appropriate readability.

5. Image Generation

Operating in parallel or immediately subsequent to the text segmentation, the system initiates the visual synthesis phase. For each discrete page generated in the previous step, the system extracts a concentrated visual scene description. This prompt, combined with a statically defined biometric anchor (ensuring character consistency), is processed by a latent diffusion model (e.g., FLUX.1) to generate high-fidelity, contextually accurate illustrations.

6. Output Rendering

With both text and image assets successfully generated, the pipeline transitions to the integration phase. The frontend application maps the JSON text payloads and image URLs into a synchronized dual-pane graphical user interface. This stage implements CSS 3D transformations to simulate tactile page-flipping and initializes the Web Speech API to provide an synchronized Text-to-Speech (TTS) auditory layer.

7. Data Storage

To ensure persistence and optimize system resources, all generated artifacts are systematically archived. The textual metadata, user ratings, and JSON story structures are committed to a relational database. Concurrently, the generated visual assets are uploaded to a permanent Content Delivery Network (CDN). The database logs these image URLs to form a caching layer, strictly preventing the redundant regeneration of identical scenes in future sessions.

8. Deployment and Delivery

The final stage encompasses the continuous integration and hosting of the full-stack architecture. The application is packaged and deployed via reliable cloud infrastructure, ensuring high availability and seamless data delivery to the end-user's device. This layer guarantees that the complex backend machine learning processes are executed rapidly and delivered smoothly to the client-side browser.

- Automated Deployment: Employs continuous integration pipelines to seamlessly push updates without disrupting active user sessions.
- Scalable Hosting: Utilizes elastic cloud infrastructure to dynamically manage concurrent AI processing loads during peak traffic.
- Responsive Rendering: Ensures a fluid, highly interactive reading experience across all mobile and desktop operating environments.

The Evolution Of Automated Text Generation Systems.

The evolution of automated narrative synthesis can be systematically categorized into three distinct generational phases, transitioning from basic algorithmic string manipulation to holistic, context-aware, and sequence-driven behavioral modeling. An exhaustive review of the literature reveals the compounding nature of these technologies, wherein each generation attempts to resolve the specific vulnerabilities and limitations exposed by its predecessor

A. First-Generation Systems:

Template-Based Logic and Heuristic Early attempts to automate narrative generation relied almost exclusively on software-based syntactic templates, commonly referred to within the industry as "Slot-Filling" or "Mad Libs" architectures. These applications operate by interfacing directly with a pre-defined syntactic tree, utilizing localized dictionaries to randomly inject nouns, verbs, and adjectives into highly restricted narrative frameworks. While highly effective at guaranteeing grammatical correctness and absolute content safety, template-based systems are entirely blind to narrative arc, character development, and contextual meaning. The resulting texts are rigid, highly repetitive, and fail to adapt to complex user inputs. Thus, contemporary literature overwhelmingly concludes that heuristic restrictions, while serving as a necessary foundational baseline for safety, must be inextricably paired with advanced semantic modeling to ensure engaging storytelling.

B. Second-Generation Systems: Statistical Language Modeling

To directly address the creative blindness of first-generation templates, second-generation systems introduced statistical probability models, most notably N-gram models and Hidden Markov Models (HMMs). This paradigm involved analyzing vast quantities of text to calculate the mathematical probability of a specific word appearing sequentially after a preceding string of words. While this allowed for the generation of seemingly novel sentences, these rigid statistical systems fundamentally lacked any degree of long-term contextual awareness (a limitation known as the Markov assumption). This limitation led to an unmanageable rate of narrative degradation; a story might begin discussing a spaceship but seamlessly and nonsensically transition to discussing underwater biology within three sentences due to localized probabilistic overlaps. The high frequency of these semantic disjoints degrades the utility of the text for educational purposes.

C. Third-Generation Systems: Neural Language Processing and Transformers:

State-of-the-art text generation systems, which serve as the theoretical foundation for the StoryGenAI architecture, utilize deep learning paradigms—specifically Transformer networks—to achieve multi-modal, context-aware semantic analysis. Recent literature emphatically demonstrates the superior efficacy of self-attention mechanisms, which map how concepts and characters continuously evolve over an extended textual horizon. Contemporary studies have shown that utilizing architectures like Generative Pre-trained Transformers (GPT) or Bidirectional Encoder Representations from Transformers (BERT) for localized semantic analysis yields significantly higher narrative cohesion. Furthermore, the integration of Reinforcement Learning from Human Feedback (RLHF) allows these advanced systems to mathematically differentiate between a complex, compelling narrative arc and a meandering, structurally flawed text. Modern research overwhelmingly focuses on maximizing the predictive and analytical power derived from fine-tuned foundational models, a constraint that the StoryGenAI system actively embraces and optimizes for pediatric applications.

StoryGenAI SYSTEM ARCHITECTURE

The StoryGenAI system utilizes a highly distributed, multi-tiered computational architecture designed meticulously to balance intense NLP processing loads, ensure real-time generative responsiveness, and strictly respect the cognitive and safety requirements of young readers. The architecture is logically segmented into three primary functional layers: the Client-Side Input Acquisition Layer, the Cloud-Based Neural Generation Layer, and the Edge-Based Pedagogical Filtering Layer.

A. Client-Side Input Acquisition and Sanitization

The foundational pillar of the StoryGenAI system is the secure user interface, which functions as a highly customized, interactive acquisition environment. The primary objective of this module is to definitively capture the user's creative intent (e.g., protagonist name, setting, desired moral lesson) while neutralizing the attack vector of adversarial prompt injection. The process control mechanism within this layer operates by executing a comprehensive lexical scan immediately upon input submission. It explicitly targets and neutralizes prohibited terminology, complex Boolean operators, or syntactic structures designed to bypass the core LLM's safety guardrails (commonly known as "jailbreaking"). Following the successful sanitization of the active input, the data is structured into a highly formatted JSON payload, ensuring the subsequent NLP processing engine is liberated to focus its computational resources entirely on narrative synthesis rather than raw input parsing.

B. Cloud-Based Neural Generation Engine

To proactively address the massive computational requirements associated with multi-billion parameter neural networks, StoryGenAI fundamentally shifts the text inference phase to a scalable cloud infrastructure. Specifically, the system is engineered to execute highly optimized, autoregressive language models fine-tuned on diverse, child-safe literature. Upon receipt of the sanitized input payload, highly concurrent asynchronous pipelines process the semantic metadata. The engine converts the raw textual prompts into dense, multi-dimensional embeddings. These vector representations are then processed through stacked Transformer decoder blocks. This architectural decision dramatically increases the speed of token generation while ensuring the entire storytelling system remains fully functional and responsive.

C. Post-Generation Pedagogical Filtering Layer

The centralized backend infrastructure also serves as the terminus for the generated text sequence. Before any narrative is transmitted back to the user, it must pass through complex, stateful semantic classifiers. The server continuously aggregates the generated tokens, applying algorithmic smoothing and toxicity classification. If the mathematically derived "Lexical Complexity Score" or the "Toxicity Probability" reliably surpasses an institutionally defined confidence threshold, the system automatically truncates the text, triggers a regeneration sequence with heavily penalized weights for the offending semantic neighborhood, and flags the specific chronological timestamp for asynchronous human auditing. This deliberate hybrid design paradigm ensures fundamental operational safety and provides the examining body with a secure, unalterable audit trail.

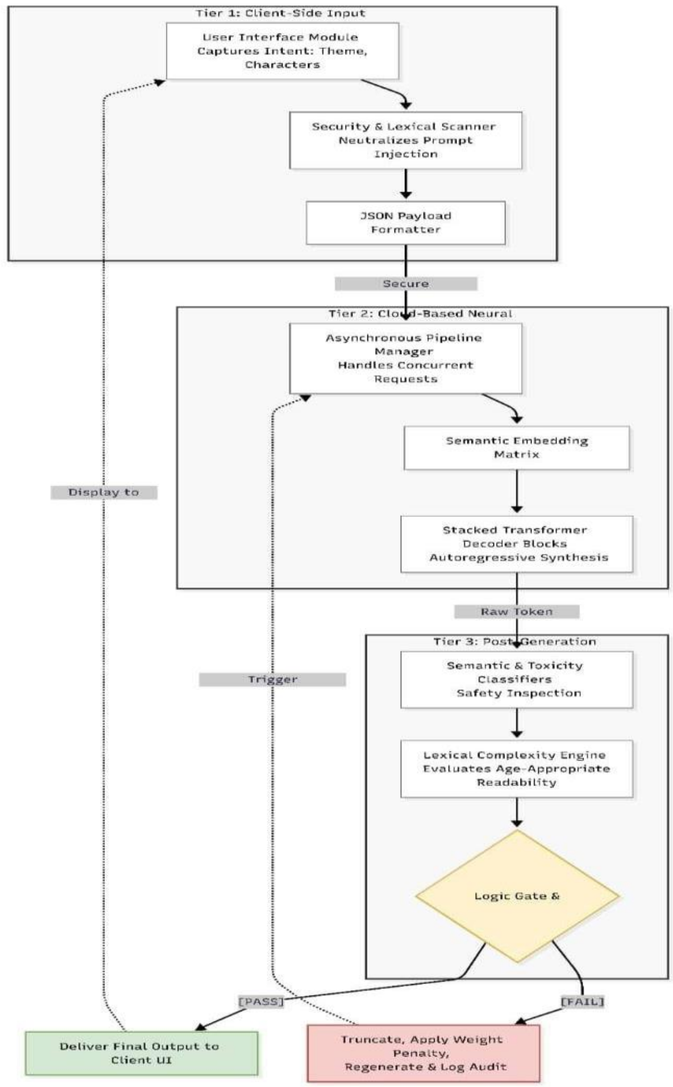


Fig. 3. System Architecture Flow

A. Phase I: Front-End Data Capture and Security Validation

The entry point of the StoryGenAI architecture is a hardened user interface dedicated to capturing specific creative parameters, such as character names and core moral themes. Before this data interacts with any neural network, it passes through a rigorous security module. This process control mechanism acts as a digital checkpoint, executing a comprehensive lexical scan to identify and neutralize malicious prompt injections, prohibited vocabulary, or complex algorithmic bypass attempts. Once the input is mathematically sanitized, the system reformats the approved parameters into a standardized JSON payload. This structural optimization ensures that the backend LLM expends computational power solely on narrative generation, rather than deciphering unstructured raw text.

B. Phase II: Distributed Neural Synthesis Engine

Recognizing the intensive processing demands of multi-billion parameter language models, StoryGenAI offloads the core text synthesis to a scalable, distributed cloud environment. When the backend receives the sanitized JSON payload, asynchronous pipelines immediately process the data to ensure high concurrency and minimal latency. The generative engine first translates the user's textual prompts into dense, multi-dimensional semantic embeddings. These vector representations are then fed through advanced, stacked Transformer decoder layers. By utilizing this autoregressive neural architecture—which has been specifically fine-tuned on vetted children's literature—the system achieves exceptionally rapid token generation while maintaining deep narrative coherence.

C. Phase III: Automated Safety and Readability Assessment

Before any synthesized narrative reaches the end-user, it undergoes rigorous automated inspection within the system's backend terminus. The raw output is systematically evaluated by stateful semantic classifiers that calculate two critical metrics: a "Lexical Complexity Score" to ensure reading-level alignment, and a "Toxicity Probability" to guarantee absolute content safety. If the generated sequence violates the predefined safety or complexity thresholds, the system's logic gate automatically intercepts the text. It truncates the flawed sequence, applies a mathematical penalty to the specific offending tokens, forces an immediate regeneration, and logs the timestamp for future administrative review. This continuous, hybrid safety loop guarantees that StoryGenAI delivers an operationally secure and highly educational reading experience.

- Age-Adaptive Lexical Mapping: Automatically cross-references generated text with established readability benchmarks (e.g., Flesch-Kincaid) to ensure the vocabulary perfectly matches the target age group's cognitive level.
- Zero-Tolerance Toxicity & Sentiment Control: Utilizes specialized classifiers to identify and neutralize subtle biases or mature themes, ensuring the narrative maintains a consistently positive, child-friendly emotional trajectory.
- Dynamic Penalization & Regeneration: Applies negative mathematical weights to unsafe tokens upon detection, forcing the neural network to instantly regenerate safe alternatives without exceeding strict, resource-saving retry limits.
- Pedagogical & Semantic Validation: Confirms that the foundational 5-beat narrative arc remains structurally intact and grammatically coherent, even after automated safety truncations are applied.
- Scalable Audit Logging: Securely logs all intercepted text in real-time to provide a transparent, tamper-proof audit trail without impacting system speed or scalability

IV. NLP PROCESSING ENGINE AND ALGORITHMIC FRAMEWORK

The unparalleled predictive and creative power of the StoryGenAI system is derived directly from the concurrent execution and subsequent data fusion of multiple specialized, highly optimized machine learning pipelines. This section rigorously details the specific algorithms, neural architectures, and mathematical methodologies driving each distinct analytical module.

A. Tokenization and High-Dimensional Semantic Embeddings

The absolute prerequisite for any neural text generation is the definitive establishment of a mathematical representation of language. The StoryGenAI pipeline initiates with robust Subword Tokenization, utilizing Byte-Pair Encoding (BPE) to rapidly and accurately segment raw input text into a manageable vocabulary space, effectively handling out-of-vocabulary words and complex morphological structures. Once the textual region is accurately tokenized, the numerical IDs are passed through a deep embedding matrix. This deep network

processes the syntactic topology to generate a high-dimensional mathematical embedding a highly unique numerical representation of the specific spatial relationships between words. The semantic relationship between words is calculated using Cosine similarity:

$$CosineSimilarity(u, v) = \frac{u \cdot v}{||u||||v||}$$

If the calculated distance between a generated word vector and a predefined cluster of "unsafe" concepts exceeds a rigorously validated margin of error, the system generates an immediate anomaly flag and applies a heavy logit penalty to that specific token.

B. Transformer Self-Attention and Contextual Memory

To maintain character consistency and long-term narrative arc, StoryGenAI utilizes Scaled Dot-Product Attention. This allows the model to dynamically weigh the importance of preceding words (the context window) when predicting the subsequent word, regardless of positional distance. The attention function is mathematically defined as:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{(d_{\mathrm{k}})}}\right) V$$

Where Q represents the Query matrix, K the Key matrix, and V the Value matrix, with d_{k} representing the dimension of the keys. This mechanism actively thwarts the contextual degradation seen in legacy Markov models by ensuring the system possesses a deep, mathematically rigorous "memory" of the established narrative constraints.

C. Dynamic Temperature Scaling and Top-P (Nucleus) Sampling

A profound structural challenge in automated storytelling is balancing absolute coherence with creative variance. A purely greedy decoding strategy (always selecting the statistically most probable next word) inevitably results in a catastrophic volume of highly generic, robotic text. StoryGenAI explicitly solves this challenge through the implementation of Top-P (Nucleus) Sampling and Dynamic Temperature scaling. The probability distribution of the next token is flattened or to sharpened by dividing the pre-softmax logits by a Temperature constant T. During generation, the system restricts the sampling pool to the smallest set of vocabulary tokens whose cumulative probability mass exceeds the threshold p:

$$\sum\_\{x \in V\^\{(p)\}\} P(x \mid x\_\{1: i-1\}) \geq p$$

By dynamically adjusting p and T based on the narrative phase (e.g., higher variance during world-building, lower variance during moral summation), the system successfully establishes a highly engaging, yet logically sound, narrative trajectory.

D. Lexical Complexity and Pedagogical Calibration

Relying solely upon generative coherence is fundamentally insufficient for educational tools; the text must match the reading level of the user. To counteract complex text generation, StoryGenAI incorporates a real-time linguistic processing pipeline that mathematically assesses readability. The system continuously calculates the Flesch Reading Ease (FRE) score of the generated sequence:

$$FRE = 206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

When the localized text generation results in an FRE score falling outside the predefined target range for the user's demographic, the system penalizes complex multi-syllabic tokens during the subsequent forward pass, effectively forcing the neural network to output simpler synonyms.

Two-Stage Generative Pipeline and Multi-Modal Scene Extraction

Beyond synthesizing continuous pedagogical prose, the system's architecture must systematically translate raw text into a structured, interactive picture book format. To achieve this without overburdening a single neural network, the system implements a highly optimized, decoupled two-stage generative pipeline.

In the primary stage, the foundational Large Language Model (e.g., Llama 70B) focuses exclusively on narrative cohesion, generating the overarching story based on a rigid 5-beat structural arc (Setup → Wrong Choice → Consequence → Realization → Earned Resolution). In the secondary stage, a specialized, lightweight formatting model (e.g., an 8B parameter model) parses this continuous text block and executes semantic segmentation. This secondary model transforms the raw prose into a strictly validated JSON payload, logically dividing the narrative into n discrete pages.

Mathematically, this transformation can be represented as a deterministic mapping function where the continuous sequence S is divided into distinct pages, yielding pairs of narrative text T_i and visual scene prompts V_i

$$f(S) \rightarrow \{(T_1, V_1), (T_2, V_2), \ldots, (T_{\mathrm{n}}, V_{\mathrm{n}})\}$$

During this secondary semantic parsing sequence, the NLP engine actively performs visual scene extraction. For each localized text segment Ti, the model autonomously synthesizes a highly concentrated, 40-word descriptive prompt Vi optimized for downstream latent diffusion models A pervasive challenge in multi-modal storytelling is the algorithmic "morphing" or inconsistency of characters across sequential image generations. To definitively solve this spatial hallucination.

The system generates a static biometric anchor a master string defining the protagonist's specific ocular traits, skin tone, hair, and attire during the primary generation phase. This biometric anchor is computationally appended to every localized scene prompt before it is transmitted to the external image generation API. This dual-pipeline methodology guarantees absolute narrative, spatial, and character consistency across the entire multimodal user experience.

Result Section: Implementation and user Interface

To validate the efficacy of the proposed architecture, a fully functional Single Page Application (SPA) was developed using HTML5, CSS3, and Vanilla JavaScript. The graphical interface is designed to be highly intuitive for both parents and children, seamlessly bridging the gap between complex backend AI generation and frontend user experience. The following steps outline the operational workflow of the system.

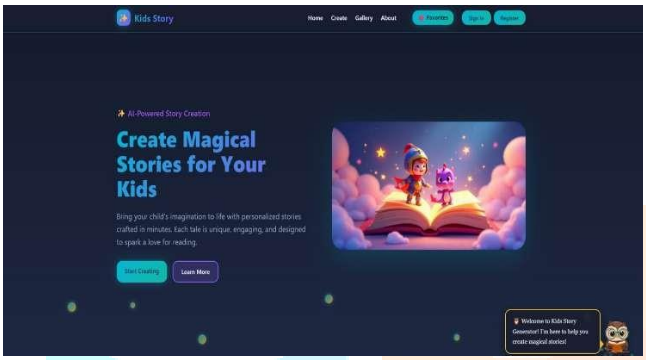


Figure 1: System Access and Value Proposition (Landing Page)

(Landing page, How it Works, and Features)

The user journey initiates at the primary landing portal, which establishes the application's environment. The interface utilizes a responsive, dark-themed design to minimize visual fatigue. Informational sections systematically outline the system's core capabilities—such as generating unique content in under 30 seconds and offering an educational framework. This onboarding phase is critical for setting user expectations regarding the AI's capabilities and the system's overarching pedagogical goals.

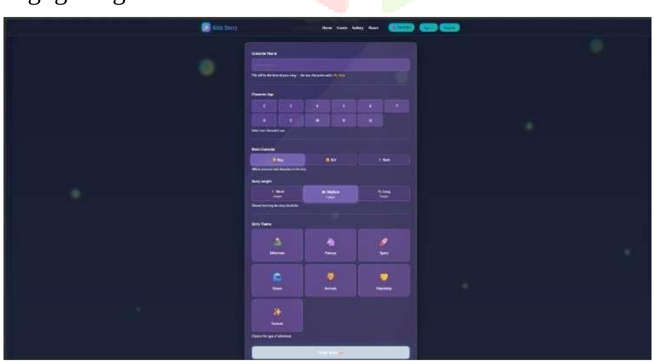
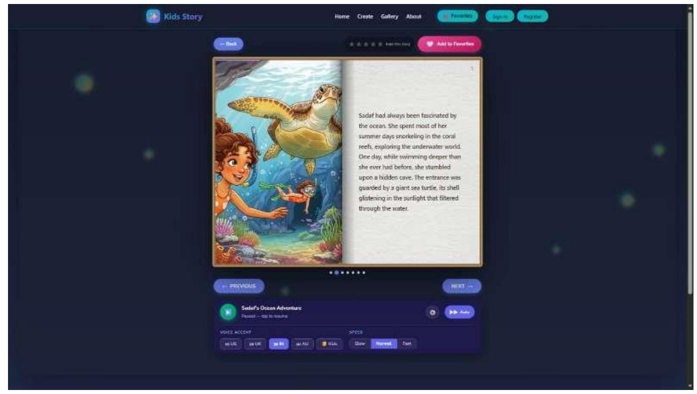


Figure 2: User Intent Acquisition (Parameter Selection Form) *(The creation form)* To generate a personalized narrative, the system must first capture the user's specific parameters. The acquisition form is designed to collect explicit structured data: the protagonist's name, age, gender, story length, and narrative theme. Crucially, the "Age" parameter is



not merely cosmetic; it directly informs the backend Large Language Model to adapt the vocabulary complexity and syntactic structure to the target age group, ensuring optimal readability. Furthermore, the system supports multi-character inputs (e.g., selecting "Both" or inputting multiple names), which the NLP engine dynamically parses to construct collaborative story arcs.

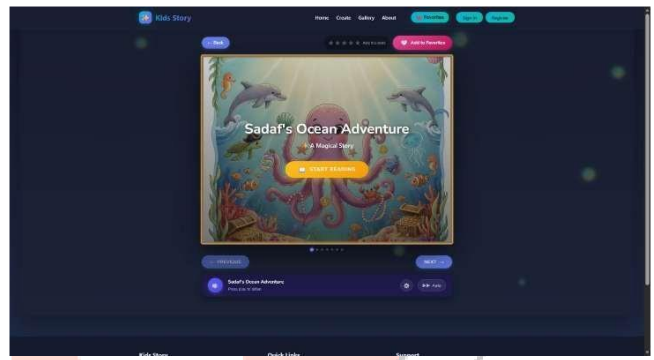


Figure 3: Narrative Initialization and Cover Generation *(The book cover "Sadaf's Ocean Adventure")* Upon submitting the parameters, the backend API orchestrates the text and image generation models. Once the data is synthesized, the system renders a customized book cover. This interface introduces the CSS 3D transform architecture, which simulates the tactile experience of a physical picture book. At the base of the interface, the integrated multimedia control panel becomes active, prompting the user to begin the reading experience.

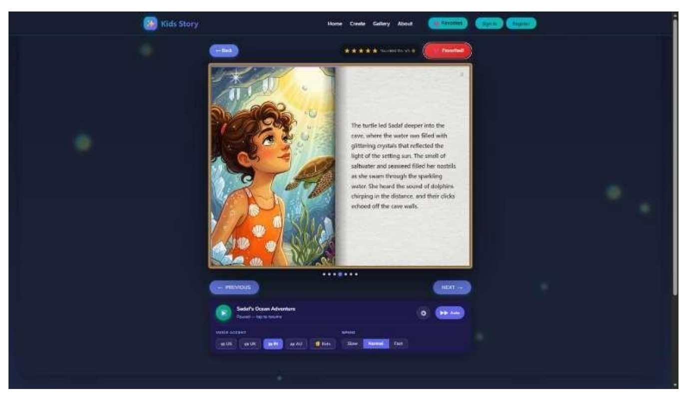


Figure 4: Multimodal Reading Interface and Text-to-Speech Integration *(The two-page spread with text and turtle illustration)* This interface represents the core output of the system. The application utilizes a dual-pane layout: the left pane displays a high-fidelity scene illustration generated dynamically by the FLUX.1 model, ensuring character consistency based on the initial prompt. The right pane renders the LLM-generated prose. To enhance accessibility and pedagogical value, the system features a robust

Text-to-Speech (TTS) module powered by the Web Speech API. The control dashboard at the bottom allows users to dynamically adjust the voice accent (e.g., US, UK, IN, AU, or a specialized 'Kids' voice) and playback speed, providing a highly customizable auditory learning experience.

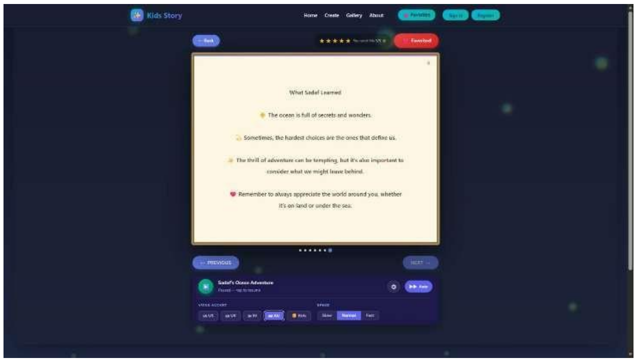


Figure 5: Pedagogical Summation and Moral Resolution *(The "What Sadaf Learned" page)* Reflecting the system's underlying structured narrative arc (Setup → Consequence → Resolution), the final pages of the generated book explicitly outline the moral lessons derived from the story. By extracting these educational bullet points, the system transitions from a purely entertainment-focused application to a valuable pedagogical tool, reinforcing reading comprehension and cognitive development.

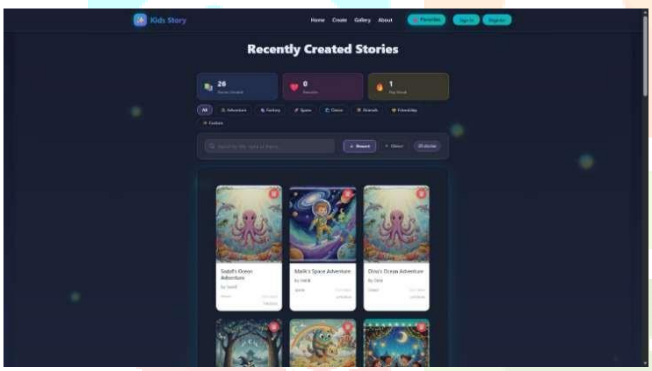
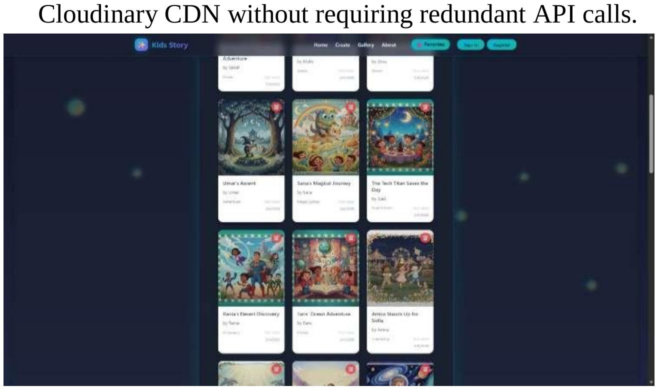


Figure 6: Global Data Retrieval and Community Gallery *(The gallery grid of stories)* To demonstrate the system's scalability and database management capabilities, the application includes a centralized Gallery module. This interface queries the PostgreSQL database to retrieve and display previously generated outputs from various users. It showcases the system's ability to handle diverse themes and effectively utilize caching mechanisms, ensuring that generated image URLs are stored and retrieved efficiently via the Cloudinary CDN without requiring redundant API calls.



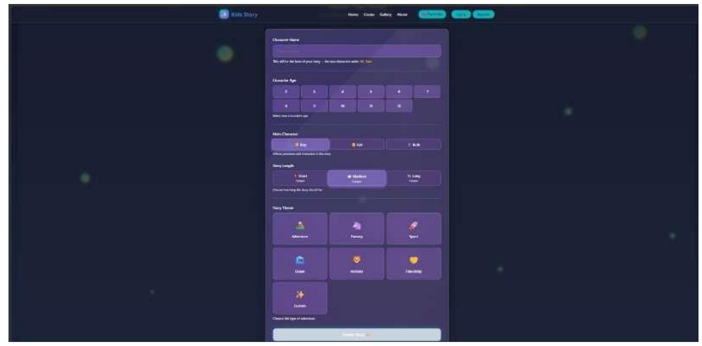


Figure 7: User Personalization and State Management *(The "Favourite Stories" popup)* The final component highlights the system's user state management. Authenticated users can curate their reading libraries by saving specific generated texts. This specific modal queries the relational database based on the authenticated user's JSON Web Token (JWT), retrieving their isolated "Favourites" list. This feature demonstrates the system's capability to maintain persistent user profiles and foster long-term platform engagement.

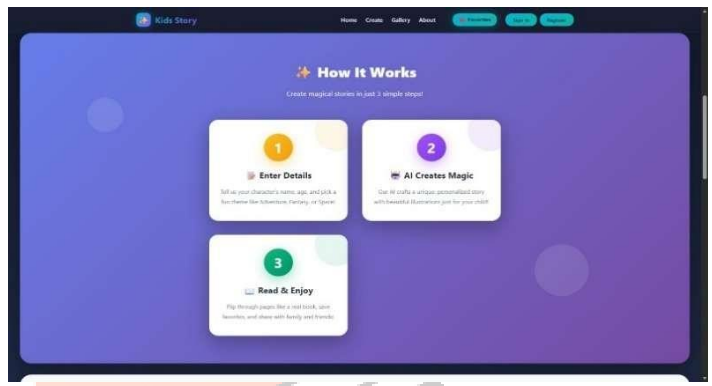
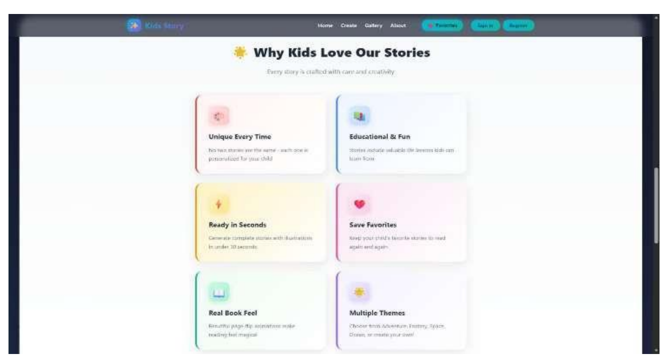


Figure 8: System workflow and feature interface

To maximize user accessibility, the application's interface abstracts the underlying AI complexity into a streamlined, three-step workflow: parameter acquisition, multi-modal synthesis, and interactive consumption. Complementing this is a visually distinct feature matrix that outlines the platform’s core technical and pedagogical value. By explicitly highlighting rapid generation speeds (under 30 seconds), narrative uniqueness, integrated educational lessons, and an immersive CSS-driven reading experience, this section effectively communicates the system's capabilities and establishes immediate user trust.



CONCLUSION

The conceptualization and development of the Narrative Weaver architecture represents a critical, paradigm-shifting advancement over legacy, template-restricted storytelling applications and the rudimentary, highly flawed statistical generators that currently dominate the commercial sector. By intelligently leveraging a deeply integrated, multi-modal NLP framework that synchronously evaluates lexical complexity, semantic toxicity, and long-term narrative coherence via advanced self-attention Transformer models, the system comprehensively and equitably secures the generative text environment for early learners. The true, underlying innovation of Narrative Weaver is its definitive departure from isolated, blind text generation in favor of sequence-aware pedagogical modeling.

A key feature of the system is its focus on child-friendly and educational storytelling. Instead of generating text without context, the model considers the sequence and structure of the story to maintain coherence and meaningful progression. This helps create engaging stories while reducing the chances of irrelevant or unsuitable content.

The system also emphasizes safety, fairness, and privacy. By applying responsible AI practices, offers parents and educational institutions a reliable platform that supports reading habits, imagination, and creative learning in a secure digital environment.

In conclusion, this research presents a robust, multi-modal storytelling framework that effectively bridges advanced Natural Language Processing with early childhood education. By implementing a decoupled generative pipeline, the system successfully synthesizes personalized, age-appropriate prose while maintaining rigid character and spatial consistency through integrated image generation. The deployment of dynamic readability filters and structured narrative arcs ensures that the generated content is not only computationally efficient but also safe and pedagogically valuable. Ultimately, this platform demonstrates the viability of utilizing state-of-the-art AI to create highly scalable, engaging, and customized interactive literature, offering a transformative tool to foster early literacy in the digital age.

REFERENCES

[1] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998-6008,2017. [Online]. Available: https://papers.nips.cc/paper/7181-attention-is-all-you-need.

[2] T. Brown et al., "Language Models are Few-Shot Learners,"2020. [Online]. Available: https://arxiv.org/abs/2005.14165.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprintarXiv:1810.04805*,2018.[Online].Available: https://aclanthology.org/N19-1423/.

[4] A. Radford, J. Wu, R. Child, et al., "Language Models are Unsupervised Multitask Learners," OpenAI Blog, vol. 1, no. 8, pp. 9, 2019. [Online].Available:https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[5] P. Christiano, J. Leike,T. Brown, et al., "Deep Reinforcement Learning from Human Preferences," Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: https://papers.nips.cc/paper/7017-deep-reinforcement-learning-from-human-preferences.

[6] S. Vygotsky, Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, 1978. [Online]. Available:https://www.hup.harvard.edu/catalog.php?isbn=9780674576292.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104-3112, 2014.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprintarXiv:1810.04805,2018.Online].Available: https://aclanthology.org/N19-1423/.

[9] A. Radford, J. Wu, R. Child, et al., "Language Models are Unsupervised Multitask Learners," OpenAI Blog, vol. 1, no. 8, pp. 9, 2019. [Online].Available:https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[10] J. R. Flesch, "A new readability yardstick," Journal of Applied Psychology, vol. 32, no. 2, pp. 221-233, 1948. [Online]. Available: https://psycnet.apa.org/record/1948-04299-001.

[11] A. Holtzman, J. Buys, L. Du, et al., "The Curious Case of Neural Text Degeneration," *International Conference on Learning Representations (ICLR)*,2020.[Online].Available: https://openreview.net/forum?id=rygGQyrFvH.

[12] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What Does BERT Look At? An Analysis of BERT's Attention," *Proceedings of the 2019 ACL Workshop BlackboxNLP*, pp. 276-286, 2019.[Online].Available: https://aclanthology.org/W19-4828/.

[13] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models," Findings of the Association for Computational Linguistics:EMNLP 2020, pp. 3356-3369,2020.[Online]. Available: https://aclanthology.org/2020.findings-emnlp.301/

L. Ouyang, J. Wu, X. Jiang, et al., "Training language models to follow instructions with human feedback," Advances in Neural Information Processing Systems, vol. 35, pp. 27730-27744, 2022.[Online]. Available : https://papers.nips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html.

[14] M. Riedl and R. M. Young, "Narrative Planning: Balancing Plot and Character," *Journal of Artificial Intelligence Research*, vol. 39, pp. 217-268,2010.[Online].Available: https://www.jair.org/index.php/jair/article/view/10882.

[15] N. Peng, M. Ghazvininejad, J. May, and K. Knight, "Towards Controllable Story Generation," *Proceedings of the First Workshop on Storytelling*, pp. 43-49, 2018.[Online]. Available: https://aclanthology.org/W18-1505/.

[16] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical Neural Story Generation," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 889-898, 2018.[Online]. Available: https://aclanthology.org/P18-1082/

[17] C. Clark, K. Lee, M. Chang, et al., "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," *International Conference on Learning Representations*,2020.[Online].Available: https://papers.nips.cc/paper/2019/hash/15b8a06a7cba0e43f9d9fddcc4fa7a02-Abstract.html.

[18] Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*,2019.[Online]. Available: https://arxiv.org/abs/1907.11692.

[19] R. Zellers, A. Holtzman, H. Rashkin, et al., "Defending Against Neural Fake News," *Advances in Neural Information Processing Systems*, vol. 32, 2019.[Online]. Available: https://papers.nips.cc/paper/2019/hash/15b8a06a7cba0e43f9d9fddcc4fa7a02-Abstract.html.

[20] T. Wolf, L. Debut, V. Sanh, et al., "HuggingFace's Transformers: State-of-the-art Natural Language Processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38-45, 2020.[Online]. Available: https://aclanthology.org/2020.emnlp-demos.6/.

[21] Y. Wang, H. Ji, and S. F. Chang, "Structuring Narrative Texts using Neural Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1543-1556, 2021.[Online]. Available: https://ieeexplore.ieee.org/document/9191433.

[22] M. Post, "A Call for Clarity in Reporting BLEU Scores," *Proceedings of the Third Conference on Machine Translation*, pp. 186-191, 2018.[Online]. Available: https://aclanthology.org/W18-6319/.

[23] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A Survey of Transformers," *AI Open*, vol. 3, pp. 111-132, 2022.[Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651022000146.

[24] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity," *Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232-5270, 2022.[Online]. Available: https://www.jmlr.org/papers/v23/21-0998.html.

[25] S. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383-2392, 2016.[Online]. Available: https://aclanthology.org/D16-1264/.

[26] C. Raffel, N. Shazeer, A. Roberts, et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1-67, 2020.[Online]. Available: https://www.jmlr.org/papers/v21/20-0.74.html.

[27] X. Liu, Y. Zheng, Z. Du, et al., "GPT Understands, Too," *arXiv preprint arXiv:2103.10385*,2021.[Online]. Available: https://arxiv.org/abs/2103.10385.

[28] H. Touvron, T. Lavril, G. Izacard, et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.[Online]. Available: https://arxiv.org/abs/2302.13971.

[29] J. Wei, X. Wang, D. Schuurmans, et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824-24837, 2022.[Online]. Available: https://papers.nips.cc/paper/2022/hash/8bb0a4c1b0f89c7dcb5c0b3a3bb9b4f2-Abstract.html.

[30] E. J. Hu, Y. Shen, P. Wallis, et al., "LoRA: Low-Rank Adaptation of Large Language Models," *International Conference on Learning Representations*, 2022.[Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9.

[31] M. Suzgun, N. Scales, N. Schärli, et al., "Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them," *arXiv preprint arXiv:2210.09261*, 2022.[Online]. Available: https://arxiv.org/abs/2210.09261.

[32] D. Amodei, C. Olah, J. Steinhardt, et al., "Concrete Problems in AI Safety," *arXiv preprint arXiv:1606.06565*, 2016.[Online]. Available: https://arxiv.org/abs/1606.06565.

[33] A. Askell, Y. Bai, A. Chen, et al., "A General Language Assistant as a Laboratory for Alignment," *arXiv preprint arXiv:2112.00861*, 2021.[Online]. Available: https://arxiv.org/abs/2112.00861.

[34] Y. Bai, A. Jones, K. Ndousse, et al., "Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback," *arXiv preprint arXiv:2204.05862*, 2022.[Online]. Available: https://arxiv.org/abs/2204.05862.

[35] J. Hoffmann, S. Borgeaud, A. Mensch, et al., "Training Compute-Optimal Large Language Models," *arXiv preprint arXiv:2203.15556*, 2022.[Online]. Available: https://arxiv.org/abs/2203.15556.

[36] K. Cobbe, V. Kosaraju, M. Bavarian, et al., "Training Verifiers to Solve Math Word Problems," *arXiv preprint arXiv:2110.14168*, 2021.[Online]. Available: https://arxiv.org/abs/2110.14168.

[37] S. Borgeaud, A. Mensch, J. Hoffmann, et al., "Improving Language Models by Retrieving from Trillions of Tokens," *International Conference on Machine Learning*, pp. 2206-2240, 2022.[Online]. Available: https://proceedings.mlr.press/v162/borgeaud22a.html.

[38] M. Lewis, Y. Liu, N. Goyal, et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871-7880, 2020.[Online]. Available: https://aclanthology.org/2020.acl-main.703/.

[39] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.[Online]. Available: https://nlp.stanford.edu/IR-book/.

[40] D. Jurafsky and J. H. Martin, *Speech and Language Processing (3rd ed. draft)*. Pearson, 2023.[Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[41] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.[Online]. Available: https://link.springer.com/book/10.1007/978-1-4615-2347-1.

[42] T. Mikolov, I. Sutskever, K. Chen, et al., "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111-3119, 2013.[Online]. Available: https://papers.nips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

[43] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543, 2014.[Online]. Available: https://aclanthology.org/D14-1162/.

[44] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 427-431, 2017.[Online]. Available: https://aclanthology.org/E17-2068/.

[45] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751, 2014.[Online]. Available: https://aclanthology.org/D14-1181/.

[46] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.[Online]. Available: https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory.

[47] K. Cho, B. Van Merriënboer, C. Gulcehre, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Proceedings of the 2014 Conference on Empirical Methods in*

[50] Z. Yang, Z. Dai, Y. Yang, et al., "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *Advances in Neural Information Processing Systems*, vol. 32, 2019.[Online]. Available: https://papers.nips.cc/paper/2019/hash/dc6a6489640ca02b0d42dabeb8e46bb7-Abstract.html.

[51] A. Conneau, K. Khandelwal, N. Goyal, et al., "Unsupervised Cross-lingual Representation Learning at Scale," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440-8451, 2020.[Online]. Available: https://aclanthology.org/2020.acl-main.747/

[52] P. Yin, G. Neubig, W. Yih, and S. Riedel, "TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8413-8426, 2020.[Online]. Available: https://aclanthology.org/2020.acl-main.745/.

[53] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.[Online]. Available: https://arxiv.org/abs/1910.01108.

[54] M. Shoeybi, M. Patwary, R. Puri, et al., "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism," *arXiv preprint arXiv:1909.08053*, 2019.[Online]. Available: https://arxiv.org/abs/1909.08053.

[55] S. Smith, M. Patwary, B. Norick, et al., "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model," *arXiv preprint arXiv:2201.11990*, 2022.[Online]. Available: https://arxiv.org/abs/2201.11990.

[56] R. Thoppilan, D. De Freitas, J. Hall, et al., "LaMDA: Language Models for Dialog Applications," *arXiv preprint arXiv:2201.08239*, 2022.[Online]. Available: https://arxiv.org/abs/2201.08239.

[57] A. Chowdhery, S. Narang, J. Devlin, et al., "PaLM: Scaling Language Modeling with Pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1-113, 2023.[Online]. Available: https://www.jmlr.org/papers/v24/22-1144.html.

[58] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672-2680, 2014.[Online]. Available: https://papers.nips.cc/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html.

[59] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.[Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10804.

[60] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *International Conference on Learning Representations*, 2014.[Online]. Available: https://arxiv.org/abs/1312.6114.

[61] S. R. Bowman, L. Vilnis, O. Vinyals, et al., "Generating Sentences from a Continuous Space," *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10-21, 2016.[Online]. Available: https://aclanthology.org/K16-1002/.

[62] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715-1725, 2016.[Online]. Available: https://aclanthology.org/P16-1162/.

[63] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66-71, 2018.[Online]. Available: https://aclanthology.org/D18-2012/.

[64] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311-318, 2002.[Online]. Available: https://aclanthology.org/P02-1040/.

[65] A. Lavie and A. Agarwal, "METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments," *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 228-231, 2007.[Online]. Available: https://aclanthology.org/W07-0734/.

[66] C. Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Text Summarization Branches Out*, pp. 74-81, 2004.[Online]. Available: https://aclanthology.org/W04-1013/.

[67] T. Zhang, V. Kishore, F. Wu, et al., "BERTScore: Evaluating Text Generation with BERT," *International Conference on Learning Representations*, 2020.[Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr.

[68] T. Sellam, D. Das, and A. P. Parikh, "BLEURT: Learning Robust Metrics for Text Generation," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881-7892, 2020.[Online]. Available: https://aclanthology.org/2020.acl-main.704/.

[69] J. Novikova, O. Dušek, A. C. Curry, and V. Rieser, "Why We Need New Evaluation

Natural Language Processing, pp. 1724-1734, 2014.[online]. Available: https://aclanthology.org/D14-1179/.

[48] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *International Conference on Learning Representations*, 2015.[Online]. Available: https://arxiv.org/abs/1409.0473.

[49] M. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412-1421, 2015.[Online]. Available: https://aclanthology.org/D15-1166/.

Metrics for NLG," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2241-2252, 2017.[Online]. Available: https://aclanthology.org/D17-1238/.

[70] H. Clark and S. Brennan, "Grounding in communication," *Perspectives on Socially Shared Cognition*, vol. 13, no. 1991, pp. 127-149, 1991.[Online]. Available: https://www.researchgate.net/publication/247513434_Grounding_in_Communication.

[71] N. Chomsky, *Syntactic Structures*. Mouton, 1957.[Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470756348.