Group 19 Project: Project LibrAIrian

Fiona Strasser FKS5726@NYU.EDU
Ada Lai AVL8891@NYU.EDU
Mia Li YL9759@NYU.EDU

Milestone 1

1. Motivation

Topic: Child-Friendly AI to Serve as Early Reading Assistant
Research question: How can an AI assistant help with critical engagement and comprehension in young readers while ensuring emotional safety, age-appropriate interaction, and limited screen dependency?
Significance: Generic AI may generate text beyond children's comprehension level, fail to restrict mature content, and discourage independent attempts at critical thinking through providing plain summary. We aim to build a chatbot that can help children learn without risking intellectual or emotional health.

2. Related Work 1: KidLM: Advancing Language Models for Children – Early Insights and Future Directions (Nayeem and Rafiei (2024))

Methods: High-quality pre-trained data was gathered through a pipeline for children's sources of identification, verification, content safety and logistical filtering. KidLM randomly masks 15% of words in input sentences. KidLM+ uses Stratified Masking, so masking probability is determined by the ratio of word occurrence in child-based vs. general corpora. Understanding was evaluated using perplexity scores on text from different grades. Stereotyping was evaluated using sentiment & toxicity scores.
Strengths: Both models had lower perplexity scores when looking at 2nd to 4th grade texts than general models, which also decreased with grade level, while most models do the reverse. They output more child-targeted language than a typical model & had better sentiment and toxicity scores than typical models (so they stereotype less).
Limitations: Limitations include lack of experimentation with stratified masking ratios, only using stereotyping to assess safety, the constrained ability to vet input documents' complexity, and the limits of an English-only model. For our project, the lack of measurement of model performance in relation to dangers like hate speech, sexual content, and violent speech is the most relevant limitation.

3. Related Work 2: Machine Assistant with Reliable Knowledge: Enhancing Student Learning via RAG-based Retrieval (Lian (2025))

Methods: Lian used a LLM+RAG system to create an educational support tool geared towards college students. They implemented a combination of sparse (BM25 algorithm) and dense (FAISS) retrieval methods for the RAG component. The results were put into

unified, then re-ranked context documents. This context was passed to the OpenAI API for relevant responses with a prompt to ensure that the LLM used an educational tone and did not answer questions outside of its knowledge range. Lian supplemented this structure by allowing users to review responses, and giving instructors a range of abilities such as tuning the LLM to include different levels of knowledge outside of the RAG database, adding new documents to the RAG database, testing the chatbot, and reviewing or correcting chatbot responses.

Strengths: The Machine Assistant with Reliable Knowledge (MARK) was able to support more students in a given amount of time than a human instructor could. It was able to answer logistical questions and explain methods for solving course problems. Additionally, its monitoring and feedback tools mean it can continuously improve and inaccuracies can be caught and addressed early.

Limitations: General LLM+RAG systems have limitations such as reliance on high quality user input material, the potential for persistent hallucinations, and the need for configuring and monitoring. MARK had limitations such as giving solutions in full instead of leaving space for student learning, only taking text input, and user hesitation when talking to a chatbot. For our project, the most pressing limitation is MARK's focus on accuracy in higher-level education, without the specific focus on proper content and tonal safety that we are concerned with for early education assistance.

4. Methodology

4.1 Planned Methods

We plan to build Project LibrAIrian using GPT-4 wrapped in a custom system that adds safety layers and educational guidance. Our approach has four main components:

4.1.1 GPT-4 with Custom Prompting

We will use the OpenAI API to access GPT-4, configuring a friendly, patient AI tutor for 6-10 year-old kids through prompts to guide the model to ask questions rather than give direct answers, use simple vocabulary, and limit conversations to 10 exchanges.

4.1.2 RAG System for Grounding

We will implement retrieval-augmented generation using Children Stories Text Corpus. We'll use sentence-transformers to embed story passages, then retrieve relevant examples when kids ask questions. This grounds responses in children's literature rather than letting the model make things up, ensuring accuracy and age-appropriate language.

4.1.3 LangChain/LangGraph Orchestration

We will use LangChain and LangGraph (from Modules 4, 9, 10) to manage the workflow. LangGraph will handle the conversation flow by checking question safety, retrieving relevant passages, generating a response, validating response safety, and sending it to the user. If something fails the safety checks, the system can try again or use a backup response.

4.1.4 Safety Filtering

We will build a multi-layer safety system that works at different points: Before GPT-4 sees the input, we'll filter out obvious inappropriate questions. After GPT-4 generates a response, we'll check readability (using Flesch-Kincaid scores via NLTK), run toxicity detection (using Hugging Face models), verify vocabulary is age-appropriate (checking against Dale-Chall word lists), and screen for prohibited topics. If anything fails, we'll regenerate with stricter guidelines.

4.2 Method Justification

RAG: Kids can't fact-check AI responses, so we need to ground everything in verified children's books. Our class (Module 5) showed us how RAG helps with hallucinations and outdated information, which are critical issues when working with children.

LangChain/LangGraph: Simple prompt engineering isn't enough for child safety. We need conditional logic to route unsafe queries differently, multiple validation steps, and the ability to retry failed generations. LangGraph (Module 10) gives us this control flow.

GPT-4: It has the best reasoning abilities for understanding context and generating thoughtful questions. Since we're focusing on teaching comprehension skills rather than just answering questions, we need a model that can understand what the child is struggling with and ask correct follow-up questions.

Embeddings for retrieval: The class (Module 5) covered how embeddings capture semantic meaning beyond keywords. Kids phrase questions in unexpected ways, so we need semantic search to find relevant passages even when they don't use the exact words from the story.

4.3 Evaluation Plan

4.3.1 Safety Evaluation

- Create a test set of 500 labeled queries (safe vs. unsafe content) and measure: sensitivity/recall (percentage of unsafe queries we catch (most critical - target $\geq$95%)), specificity (percentage of safe queries we correctly allow (target $\geq$95%)), and accuracy (overall correct classifications)

- For generated responses (1,000 samples), measure toxicity detection: sensitivity (percentage of toxic responses caught (target $\geq$95%)) and accuracy (percentage of responses meeting readability standards (target $\geq$95%))

4.3.2 User Experience

- **Response time:** Mean <5 seconds, 95th percentile <10 seconds

- **Retrieval quality:** Precision around 3 $\geq$ 70%

- **Conversation flow:** Proper context maintenance and session limits

4.3.3 Baseline Comparison

Run identical tests on vanilla GPT-4 and calculate percentage improvement, especially for safety sensitivity.

4.4 Course Tools/Technologies Used

Core LLM Stack: OpenAI API (GPT-4) - Base model (Module 4), PyTorch - Framework for custom components (Module 2), Hugging Face Transformers - Safety models and potential LoRA fine-tuning (Modules 6, 7)
RAG and Orchestration: LangChain - Prompt templates, memory, retrieval integration (Modules 4, 5), LangGraph - Workflow management and safety filtering (Module 10), sentence-transformers - Create embeddings for retrieval (Module 5)
Safety and Analysis: NLTK - Readability scores, Pandas/NumPy - Data processing, Toxicity detection models from Hugging Face (Module 8 concepts)
Development: Jupyter Notebook - Main development environment, Google Colab - Testing and prototyping, NYU HPC Cluster - For processing embeddings and datasets, Git/GitHub - Version control
Evaluation: ROUGE/BLEU - Response quality (Module 13), Matplotlib/Seaborn - Visualizations

5. Dataset

Description: We will use the Children Stories Text Corpus from Kaggle, containing public domain children's stories from Project Gutenberg. It has been processed and cleaned, with metadata and offensive language removed. A word-level recurrent neural network (LSTM) was used to capture the underlying statistics of language in children's books.

Dataset link: `https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus`

Justification of Suitability: The Children Stories Text Corpus dataset from Kaggle is well-suited for the project because it consists of linguistic patterns and narrative structures found in children-appropriate literature. We plan to analyze the books' language, focusing on readability, sentence complexity, and tone through sentiment analysis. It does not contain the full text of the children's books, but it will help us define safe and age-appropriate content for children. The dataset and its analysis will also guide our safety filters and response validation process, building a foundation for our RAG system and safety evaluation.

EDA and Visualization: The dataset contains 62 unique children's stories of varying lengths and complexities. The distribution of total sentences and of total words per story is heavily right-skewed, so most stories have a relatively small number of sentences. We then looked at average number of words per sentence, which had a roughly symmetric distribution. We plotted the 20 most common words, which were all everyday words. We then found the number of unique words to measure vocabulary diversity in each story. This histogram was heavily right-skewed, so most stories have a simpler vocabulary. We examined the relationship between the total number of words and the number of unique words in a scatterplot. This revealed a roughly

linear relationship, so stories with more words generally also contained more unique words. There were a few outliers, representing stories with either unusually repetitive language or unusually diverse vocabulary relative to their length.

6. Work Plan

Contributions So Far: Fiona did the initial literature review for topic selection. Fiona did related works, Mia did the methodology, and Ada did the dataset. Ada and Mia attended the meeting with Kerr to check in. Fiona drafted the work plan section.

Weekly Plan:

10/12 - 10/18: Prepare & embed data(Fiona); create API call to GPT-4 (Ada) with underlying prompt to guide safety and filtering (Mia)

10/19 - 10/25: Implement LangChain & LangGraph for filtering & safety checks; create logic: handle input & check if questions are safe (Fiona), retrieve passage & generate response (Mia), & validate that user response is safe (or send backup response) (Ada)

10/26 - 11/1: Create 500 labeled queries of safe versus unsafe content (done between all group members); test model on these queries; analyze results for safety (Fiona), user experience (Mia), and compare to non-augmented GPT-4 (Ada)

11/2 - 11/8: Improve aspects of model that under-perform in testing; add additional features to improve performance, especially any needed safety features (all group members, with specifics dependent on model performance in testing)

11/9 - 11/15: Analyze results to date (DRI: Mia); draft new methods and find new data (DRI: Ada); finish formal write-up and turn in Milestone 2 (DRI: Fiona)

11/16 - 11/22: Implement improvements; re-test model and analyze these results

11/23 - 11/29: Create nicer front-end for user interaction; ensure kid-friendly graphics and text; continually update areas of milestone 3 until due (methodology (DRI: Mia), results & analysis (DRI: Fiona), conclusion & workflow (DRI: Ada)) “

11/30 - 12/6: Work on final deliverables: final slides (DRI: Mia), GitHub code quality/clarity check (DRI: Fiona), & GitHub ReadMe & tutorial completion (DRI: Ada)

12/7 - 12/14: Practice and deliver presentation, complete survey, and turn in final project

Member Responsibilities: Our tentative division of tasks is specified in the parentheses next to the tasks in the weekly plan. Where DRI is specified in the parentheses, the work is to be divided among all group members, but the DRI (directly responsible individual) is in charge of ensuring that it gets completed in a timely manner. All members are responsible for assisting with each other as needed and contributing to the overall analysis.

Appendix A. EDA Figures

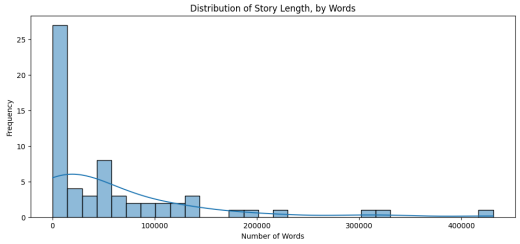

Figure 1: Histogram of Story Length, by Words

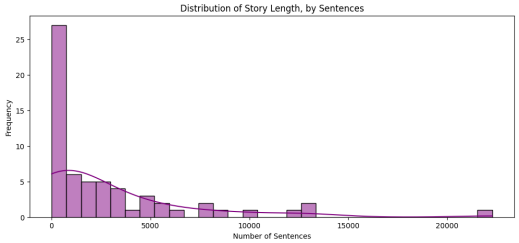

Figure 2: Histogram of Story Length, by Sentences

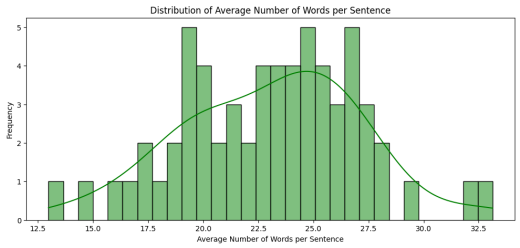

Figure 3: Histogram of Average Number of Words per Sentence

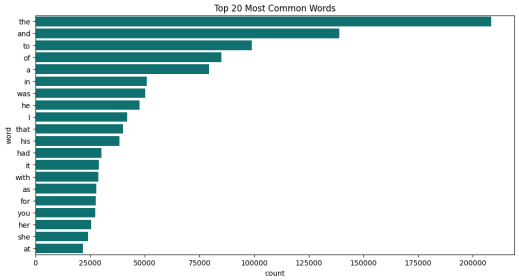

Figure 4: Bar Plot of Top 20 Most Common Words

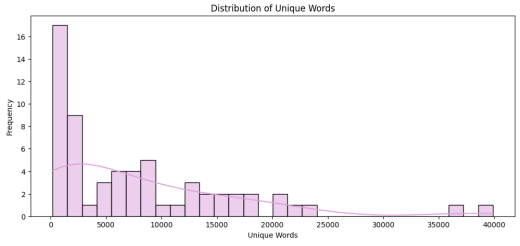

Figure 5: Histogram of Unique Words Distribution

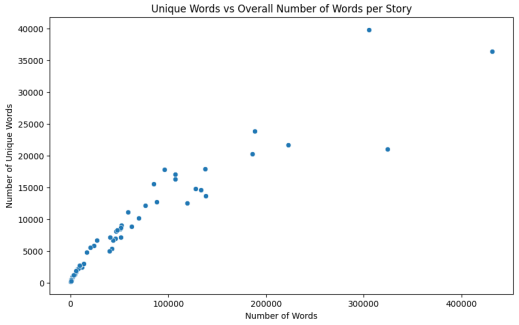

Figure 6: Total Number of Words vs. Unique Words

References

Yongsheng Lian. Machine assistant with reliable knowledge: Enhancing student learning via rag-based retrieval, 2025. URL https://arxiv.org/abs/2506.23026.

Mir Tafseer Nayeem and Davood Rafiei. KidLM: Advancing language models for children – early insights and future directions. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4813–4836, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.277. URL https://aclanthology.org/2024.emnlp-main.277/.