A Major Project Synopsis on

AI-Powered Content Generation & Speech Synthesis with RAG and TTS

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

2023-2025

by

Jagriti

23FS20MCA00023

Under the guidance of

Dr. Devershi Pallavi Bhatt

Department of Computer Applications

School of AIML, IoT&IS, CCE, DS and Computer Applications Faculty of Science, Technology and Architecture

Manipal University Jaipur Jaipur, Rajasthan

2025

I. Introduction

With the rapid advancement of Artificial Intelligence and Natural Language Processing (NLP), there has been a growing need for more efficient and human-like Text-to-Speech (TTS) systems, as well as intelligent information retrieval and generation models. This project, undertaken as part of my internship at Clipo AI, focuses on developing and optimizing AI-powered voice synthesis and content generation solutions by integrating Retrieval-Augmented Generation (RAG), Kokoro-ONNX TTS models, and Prompt Engineering techniques.

Retrieval-Augmented Generation (RAG) is an innovative approach that combines pre-trained language models with external information retrieval to enhance the quality and accuracy of generated content. This technique ensures that the AI system generates contextually relevant and factually accurate responses.

Kokoro-ONNX TTS is a high-performance, open-source text-to-speech solution that leverages the ONNX (Open Neural Network Exchange) framework to deliver fast and efficient voice synthesis. It is designed to produce natural and expressive speech outputs, making it suitable for applications such as podcast generation, virtual assistants, and automated voiceovers.

Prompt Engineering is a crucial skill in optimizing the performance of large language models (LLMs). It involves crafting precise input prompts to guide the model in generating high-quality outputs, which is essential when working with generative AI models.

During my internship at Clipo AI, I worked on deploying Kokoro-ONNX TTS models, integrating RAG-based systems for better content generation, and fine-tuning prompt engineering techniques to enhance the quality and efficiency of AI-generated outputs. This project aims to consolidate my learning and contribute to building a robust AI pipeline capable of delivering accurate and natural-sounding voice outputs from text, driven by contextually enriched content generation.

About Clipo AI

Clipo AI is an innovative startup transforming video editing and content repurposing through advanced artificial intelligence. By leveraging machine learning models, Clipo AI automates essential video creation tasks such as automatic clipping, subtitle generation, emoji and GIF suggestions, AI-powered virality predictions, and speech recognition. The platform enhances efficiency for content creators and businesses by significantly reducing manual effort and increasing editing speed. It integrates technologies like speech-to-text (ASR), text-to-speech (TTS), and Retrieval-Augmented Generation (RAG) to enable AI-driven voiceovers, intelligent content retrieval, and personalized recommendations. Clipo AI has been recognized by industry leaders through programs such as Google for Startups, Microsoft for Startups, and MongoDB for Startups, reinforcing its commitment to innovation in AI-powered video editing.

Why Choose Us?

At Clipo AI, we are revolutionizing AI-driven content generation and speech synthesis with cutting-edge technologies like Retrieval-Augmented Generation (RAG) and Kokoro-ONNX TTS. Our expertise in LLMs, prompt engineering, and efficient AI deployment ensures high-quality, context-aware, and natural-sounding outputs.

- Precision & Contextual Accuracy – RAG integration enhances factual consistency.
- Human-Like Speech Synthesis – Kokoro-ONNX TTS delivers expressive and natural voices.
- Optimized Performance – Lightweight, scalable, and efficient AI models.
- Innovation & Customization – Tailored AI solutions for diverse industry needs.

Choose us to experience next-gen AI-powered automation with the perfect blend of accuracy, efficiency, and scalability.

II. Problem Statement

AI-driven video editing and speech synthesis face challenges that impact their accuracy, efficiency, and usability. Traditional video editing is time-consuming, requiring manual effort for clipping, subtitles, and content optimization. AI-generated content often lacks contextual accuracy due to static training data, leading to inconsistencies in video summaries and recommendations.

1. Contextual Inaccuracy in AI-Generated Content
Traditional Large Language Models (LLMs) often generate hallucinated, irrelevant, or outdated information due to their reliance on static training data. This limitation affects the accuracy of AI-generated video summaries, subtitles, and recommendations. The absence of a dynamic retrieval mechanism prevents AI models from accessing real-time, factually accurate information, leading to inconsistencies in content generation.

2. Limitations in Text-to-Speech (TTS) Systems
Speech recognition models may generate inaccurate transcriptions, affecting the quality of subtitles and captions. Many existing TTS models produce robotic, monotone, or unnatural speech, making them unsuitable for professional applications such as podcasts and virtual assistants. Additionally, the high computational cost of speech synthesis models limits their accessibility, preventing seamless deployment on low-resource devices.

3. Optimization Challenges in AI Deployment
Deploying AI models in real-world applications requires balancing speed, accuracy, and computational efficiency. Generative AI models demand optimized prompt engineering techniques to ensure relevant, coherent, and context-aware outputs. Furthermore, AI-powered video processing requires lightweight and scalable architectures that can efficiently process large amounts of data without compromising quality. Ensuring AI systems operate seamlessly within content creation workflows remains a key challenge.

Project Objective

To address these limitations, this project integrates advanced AI techniques, including Retrieval-Augmented Generation (RAG) for improved contextual accuracy, Kokoro-ONNX TTS for natural speech synthesis, and AI-driven automation for video editing. By leveraging speech-to-text (ASR), optimized prompt engineering, and machine learning-based video processing, the goal is to develop a scalable, efficient, and intelligent AI-powered system that enhances video content creation while reducing manual effort.

III. Methodology/ Planning of work

The project follows a structured approach, integrating Retrieval-Augmented Generation (RAG), Kokoro-ONNX TTS, and Prompt Engineering to develop a highly efficient AI-powered content generation and speech synthesis system. The workflow is divided into multiple phases as outlined below:

Phase 1: Research & Requirement Analysis

- Understanding the limitations of existing LLM-based content generation and TTS models.
- Exploring RAG architecture to enhance contextual accuracy in generated text.
- Analysing the computational efficiency of Kokoro-ONNX TTS for real-time speech synthesis.
- Identifying optimal prompt engineering techniques to improve AI-generated outputs.

Phase 2: Data Collection & Preprocessing

- Collecting text and speech datasets for fine-tuning AI models.
- Preprocessing text data for retrieval-augmented generation (RAG) implementation.
- Preparing voice datasets to evaluate Kokoro-ONNX TTS performance.

Phase 3: Model Integration & Development

- Implementing RAG-based retrieval system to improve response accuracy.
- Integrating Kokoro-ONNX TTS for natural-sounding voice synthesis.
- Applying prompt engineering techniques to optimize LLM-generated outputs.
- Developing an efficient pipeline to connect text generation with speech synthesis.

Phase 4: Testing & Performance Evaluation

- Evaluating content quality using BLEU Score and factual consistency checks.
- Assessing speech synthesis quality through Mean Opinion Score (MOS).
- Measuring latency, response time, and computational efficiency of the system.

Phase 5: Optimization & Deployment

- Fine-tuning RAG and TTS models for scalability and real-time performance.
- Deploying the final AI-powered system for practical applications like podcasts, virtual assistants, and automated voiceovers.
- Documenting findings and creating a comprehensive project report.

IV. Requirements for proposed work

To successfully implement the AI-powered content generation and speech synthesis system, the project requires a combination of hardware, software, datasets, and tools.

1. Hardware Requirements
High-performance GPU (NVIDIA RTX 3060 or higher) – Required for model training and inference.
16GB+ RAM – Ensures smooth processing of large datasets and AI models.
High-speed SSD (512GB or more) – For efficient data storage and retrieval.
Cloud/Server Access – For scalable deployment of AI models.

2. Software Requirements
Python (3.8+) – Primary programming language for AI model development.
PyTorch / TensorFlow – Deep learning frameworks for model training.
ONNX Runtime – For optimizing Kokoro-ONNX TTS.
Hugging Face Transformers – For implementing RAG-based text generation.
FastAPI / Flask – For developing an API interface for the AI system.

3. Dataset Requirements
Pretrained LLM (e.g., GPT-3, LLaMA, or Mistral-7B) – Base model for content generation.
RAG Training Data – A collection of domain-specific knowledge bases for retrieval augmentation.
Speech Dataset (LibriSpeech / VCTK / Custom) – For evaluating and fine-tuning Kokoro-ONNX TTS.

4. Additional Tools & Libraries
LangChain – For implementing RAG-based document retrieval.
Whisper ASR – For integrating speech-to-text capabilities.
Prompt Engineering Techniques – To optimize LLM responses for accuracy and coherence.

V. Bibliography/References

1. Retrieval-Augmented Generation (RAG)
Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. [NeurIPS 2020]. Retrieved from https://arxiv.org/abs/2005.11401

2. Large Language Models & Prompt Engineering
Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. [NeurIPS 2020]. Retrieved from https://arxiv.org/abs/2005.14165
OpenAI (2022). Best Practices for Prompt Engineering with GPT-3. Retrieved from https://platform.openai.com/docs/guides/prompt-engineering

3. Text-to-Speech (TTS) & Kokoro-ONNX
Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... & Saurous, R. A. (2018). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. [ICASSP 2018]. Retrieved from https://arxiv.org/abs/1712.05884
Kokoro AI. Kokoro-ONNX: Lightweight and Efficient TTS Model. Retrieved from https://github.com/thewh1teagle/kokoro-onnx/tree/main

4. ONNX & AI Model Optimization
Microsoft. ONNX Runtime: Optimized Machine Learning Execution. Retrieved from https://onnxruntime.ai/

5. Additional Tools & Libraries
Hugging Face. Transformers Library for NLP Models. Retrieved from https://huggingface.co/docs/transformers
LangChain. Building Context-Aware AI Applications. Retrieved from https://python.langchain.com/