LLM POC Overview

Application of Large Language Models in Game Production

Author	Michael Carlos
Contributors	N/A
Approver	Michael Carlos
Date	Feb 22, 2024
Status	Approved
Security	Confidential
Document ID	Doc No./ID TBD
Location	Location TBD
Filename	LLM_POC_Overview_1.0.odt
Version	1.0

History

Version	**Date**	**Author**	**Changes**
0.1	Dec 15, 2023	Michael Carlos	Initial Draft.
0.2	Feb 18, 2024	Michael Carlos	Cleanup
1.0	Feb 22, 2024	Michael Carlos	Approved for publishing.

Table of Contents

Table of Figures

1 Vocabulary

Term	**Description**
Classifier	A neural network capable of categorizing input data.
Features	The parameters describing input data (e.g. floor area, house cost, etc). The feature count is roughly proportional to the number of nodes on the input layer of a Neural Network
FNN	Feed-forward Neural Network. A neural network where inference is calculated from the input layer to the output layer in order. Based on a Perceptron but with one or more hidden layers.
Fine-tuning	The process of additional training of a neural network with data outside of the initial corpus.
FM	Foundation Model. Neural networks that provide a base for further fine-tuning or transfer learning.
LLM	Large Language Models Needs explanation
ML	Machine Learning
Model	Any software or mathematical system that emulates a real-world physical system. In AI a model could be a neural network simulating some aspect of human intelligence.
NLP	Natural Language Processing
NN	Neural Network
POC	Proof of Concept
RAG	Retrieval-Augmented Generation. A framework that allows a user to query a pre-trained model with new information.
RL	Reinforcement Learning
Supervised Learning	Training a neural network where the desired output is presented along with the input.
Token	The unit of text data used in training or inference. The size is typically a compromise between letter-level resolution and word-level resolution based on statistical analysis of the training data.
Transformer	The name of the architecture used in some generative models. Needs explanation
Transfer Learning	A technique used to fine-tune a foundation model to perform tasks it wasn’t originally trained to do.
Unsupervised Learning	Training a neural network with unlabelled data. Normally this involves clustering data points in multidimensional space using techniques such as e.g. k-means clustering.

2 Introduction

Since the early 60s, scientists and engineers have dreamed about machines that could behave like they were alive. However, only in recent years has technology advanced to the point where this can become reality. For the first time in history, machines that can closely mimic human intelligence will proliferate through our daily lives.

Along with image, voice and motion generation, text generation using Large Language Models (LLMs) has quickly become very popular because of how well they perform in many business applications. The games industry is poised to take advantage of the many advances in the field of Artificial Intelligence.

Currently, the domain of Artificial Intelligence is very roughly divided into two areas, Rule-based systems and Machine Learning.

- Examples of Rule-Based Systems (RBS) includes
  - Unreal Engine Blueprints
  - Expert Systems
  - Fuzzy Logic
  - Finite State Machines (FSMs)
- Machine Learning (ML) covers
  - Supervised Learning (SL)
  - Unsupervised Learning (UL)
  - Reinforcement Learning (RL)

The study of Large Language Models (LLMs) currently exists as a sub-domain overlapping both Supervised Learning and Unsupervised Learning in Machine Learning.

Some of the limitations of LLMs and other AI models are:

- Knowledge cut off dates
- Censorship
- Bias
- Hallucinations
- Behaviour limitations
- Inability to learn continuously

The **challenge** addressed in this Proof of Concept (POC) is to combine all current modes of generative AI into one UI to allow researchers to easily compare and combine modes to quickly innovate new ways to overcome the limitations listed above. The goal is to create true AGI.

In addition to this, local isolated implementations potentially disconnected from the internet will be preferred to protect IP and confidential or secret data.

2.1 Document Scope

The scope of this document is LLMs and their applications within AGI Labs Inc.

2.2 Document Purpose

The purpose of this document is to provide a general base understanding about Artificial Intelligence and to describe the POC for an LLM tool for company staff. This document also serves to provide a vision and focus for the R&D team.

2.3 Audience

This document is intended for AGI Labs Inc. R&D team members. This document is not suitable for general review.

2.4 Vision

Many staff will use AI tools daily for ideation and to assist in coding and content generation. The boost in efficiency will allow us to hit scheduled launch dates with higher quality products.

2.5 Objectives

The overall objective of AI R&D is to provide staff with AI-powered solutions. The expected outcome of the POC is to check the quality and performance of LLMs in various applications. This study will allow AGI Labs Inc. to estimate the potential of LLMs as an internal tool.

3 POC High-Level Requirements

These are the high-level requirements for the POC. The word “shall” indicates a mandatory requirement and the word “should” indicates a soft requirement. [TBD Numbering may change in the draft document]

- LLM1 The POC shall combine all current AI modes in one UI.

- LLM2 Confidential data shall not be shared with external companies.
- LLM3 The tool should be designed to scale up to multiple simultaneous users. However, for the POC, a single-user environment is acceptable.
- LLM4 The POC should be securely accessible from any of the business offices.
- LLM5 The POC shall respond to queries and requests with the most up-to-date information available.
- LLM6 POC Costs should not exceed US$1k per month.
- LLM7 The POC shall log prompts and results for purposes of internal fine-tuning.
- LLM8 Post POC a version of the tool should be executable on any standard PC where possible. (i.e. push compute to the edge for portability and scalability. Internet access optional for security.)
- LLM9 The architecture shall be extensible to enable the addition of modes and models that are made available in the future. For example Fast-Text-to-Image, Text-to-Video, Text-to-3D and Gaussian Splatting.

4 Proof of Concept

This tool is strictly intended for ideation purposes. The generated content shall not be used directly in game assets without thorough review and approval by the creative staff.

4.1 User Interface

The user interface (UI) for POC will be simple at the start. The software development cycle will be iterative and end-user feedback and requests will be integrated based on business priorities. The POC will initially be web-based to allow global access. There are three main areas in the UI.

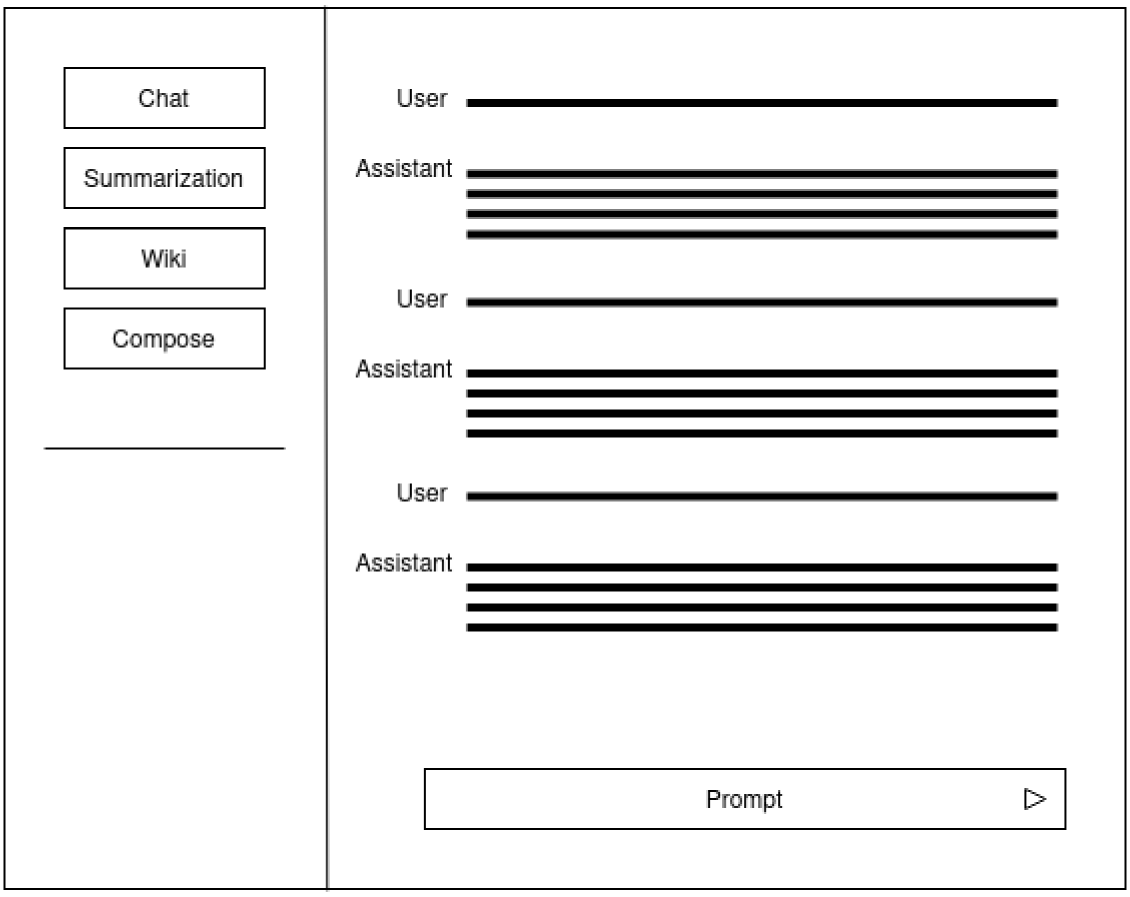

Figure 1: POC UI Wireframe

4.1.1 Modes Examples

Buttons will allow the user to select between several modes. These buttons can be placed on the top or the side.

4.1.1.1 Search (Default - DuckDuckGo Agent - General use)

A DuckDuckGo Agent will be used to fetch up-to-date information from the internet about any topic. This data will be fed into the LLM via an n-shot template and the LLM will be used to generate a coherent response. This is an alternative to fine-tuning which requires time and resources to execute. A chat type interface will be displayed and chat history will be retained for conversation context. Wikipedia and Google agents can also be used in this mode.

This mode will allow you to run creation queries (e.g. Write a poem…) on current information and recent events.

4.1.1.2 PDF (Upload document - General use)

Any short document can be uploaded (PDF format) and a summary can be requested.

4.1.1.3 Wiki / Novel Query (Vector store)

A vector store containing wiki data will be created. In this mode, every prompt will generate a query to the vector store. This vector store can be queried for specific information.

4.1.1.4 Code (Simplified Code Co-pilot or Code Whisperer)

This mode will allow the user to request a type of code snippet and specify a language. It will then generate a block of code. If the quality is acceptable, the tool may be integrated into IDEs in future projects.

4.1.1.5 Agent (Personas + context)

This mode will let you set system prompts that customize personas and context to test performance as expert agents.

4.1.1.6 Style (Fine-tuning)

This mode generates text in a specific style. Fine-tuning is used instead of RAG, so the prompt goes directly to the LLM. This will generate text in the style of e.g. a pirate. Context can also be integrated by querying a vector store. Training data includes:

- Game design documents
- Narratives
- Localized dialog
- Novels

4.1.2 Data Area

Prompts and responses will be shown in the data area. Historical data will be displayed based on the mode. In future iterations, graphics or voice generation may be integrated in the tool depending on feasibility.

4.1.3 Prompt Field

A prompt text box will be displayed in the UI. Initially it will be on the bottom.

4.2 RAG Architecture

This block diagram describes the high-level Retrieval Augmented Generation (RAG) architecture used in the initial POC.

The architecture consists of several independent sections:

- The data source that provides additional information not available in the model. This can be a database or agents e.g. Vector Databases, Google Agent, Wolfram Alpha Wrapper, etc.
- A chaining framework such as LangChain.
- A UI layer. A web framework such as Streamlit or Chainlit will be used initially.
- Large Language Models such as Llama2, Mistral or OpenAI's ChatGPT (Preference to models licensed for commercial use).

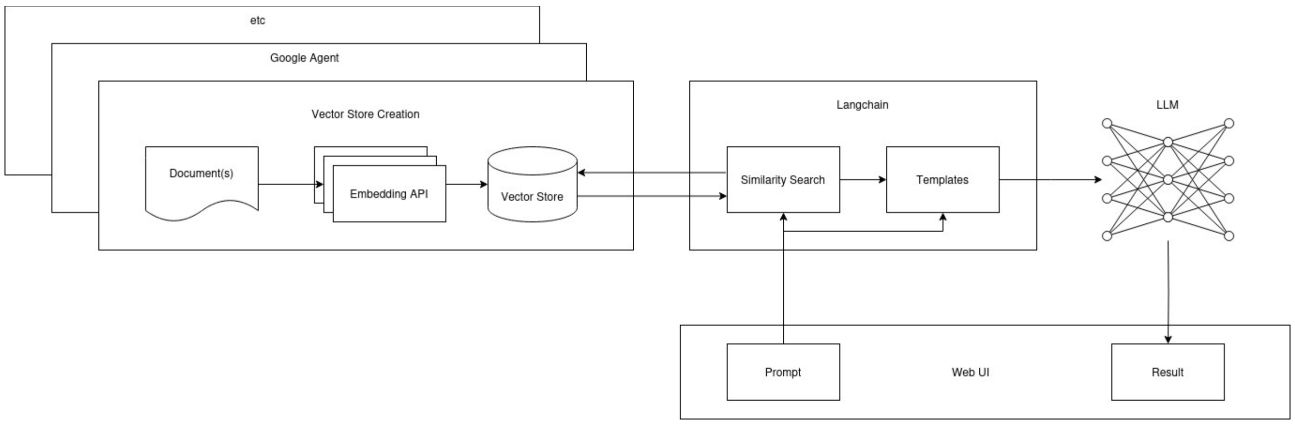

Figure 2: RAG Architecture

4.3 Sequence

This diagram clarifies the order of messages between entities. Note that the prompt is sent to the similarity search and templates simultaneously.

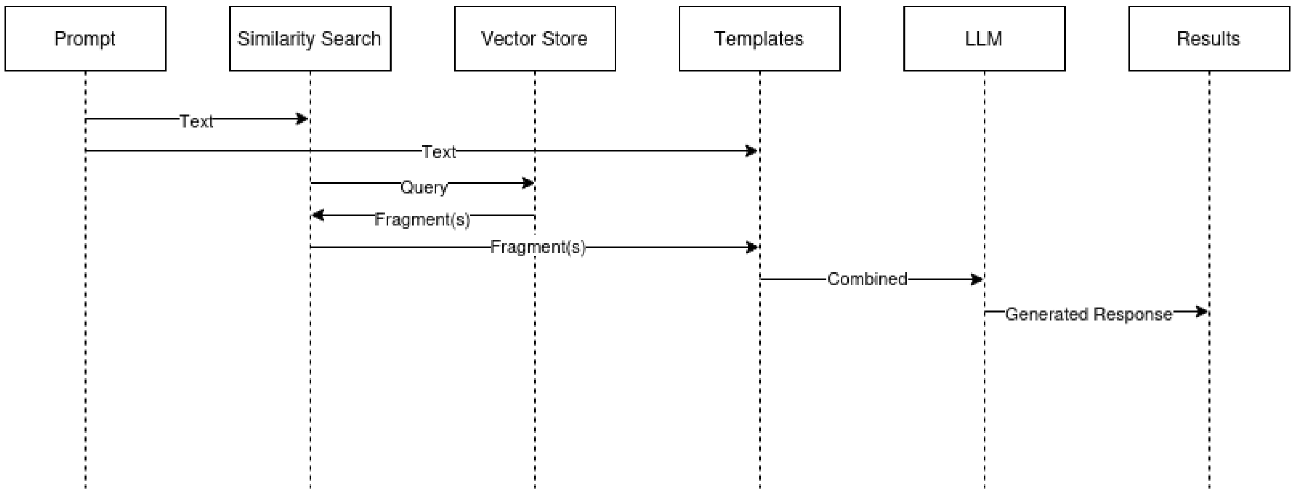

Figure 3: RAG Sequence Diagram

4.4 Deployment

The POC will be deployed on a local machine with the following minimum specifications.

- RTX 4090 24GB VRAM
- i9 latest generation or better
- 64GB RAM
- 1T SSD

5 To-do

- Chain STT–RAG–LLM–TTS
  - STT: Vosk, Whisper
  - TTS: Festival, espeak, gtts, xtts
  - Already prototyped on command line -> Transfer to UI
- RAG
  - Old implementation uses deprecated libraries -> Need to update
- Video Generation
  - Sora or next open model available
- Add functions for personas to call.