Report Here!

Cyberlife AI: A No-Code Platform for Building Memory-Enhanced Conversational Agents with RAG

Team: Brad Chen, Eagle Lo, and Webber Wu
Mentor: Adam Paulisick

Carnegie Mellon University
Master of Computational Data Science

Introduction & Problem Statement

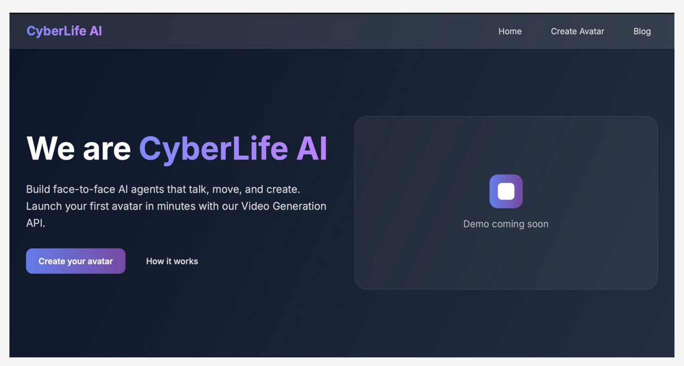
Current conversational AI platforms

- Constrain user customization (e.g., Replika, Character.AI)
- Do not maintain persistent memory across conversations
- Lack support for domain-specific or personalized knowledge

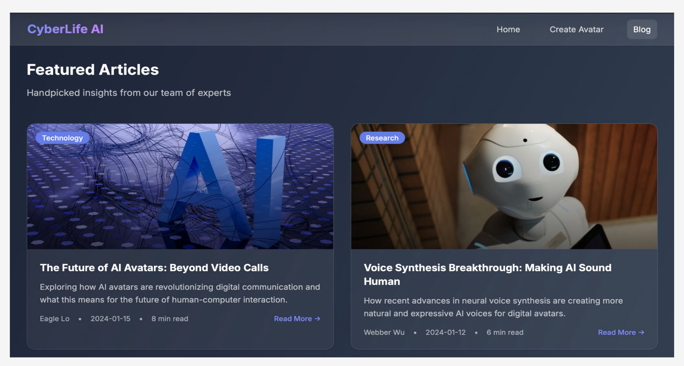
Our Solution

- Provides a no-code builder for personalized AI agents
- Implements a memory-augmented dual-retrieval RAG pipeline
- Supports user-uploaded documents and custom knowledge bases

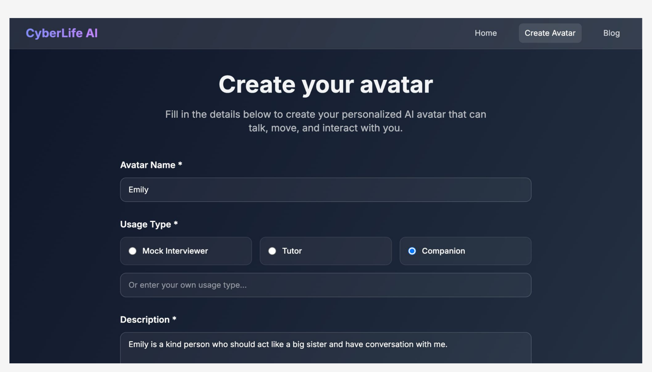
User Interface



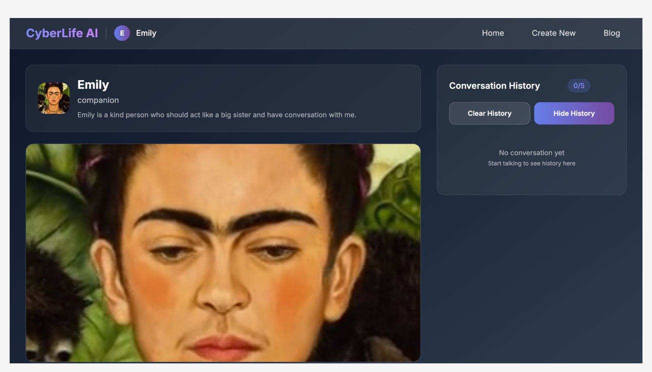
Home Page



Blog Page



Avatar Creation Page



Avatar Interaction Page

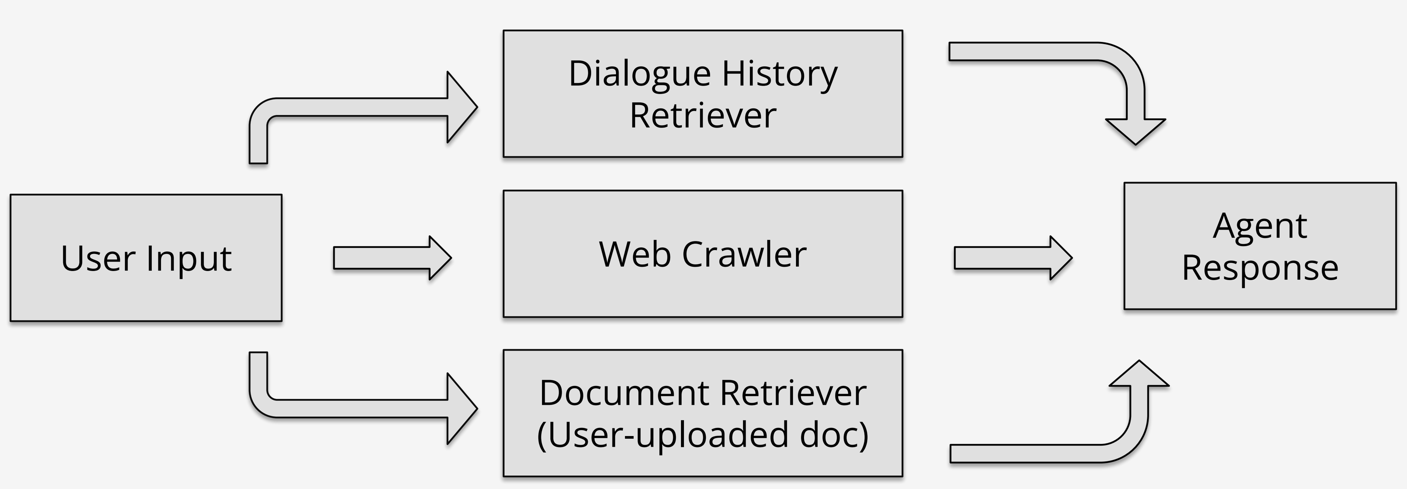
Functional Architecture

Pipeline Overview

- Input: Receives user message and agent settings
- Dual Retrieval: Fetches relevant dialogue history and documents
- Memory Module: Stores and recalls long-term user facts
- LLM Reasoning: Generates the final agent response

Requirements

- Achieves low-latency RAG retrieval
- Provides scalable storage for memory and documents
- Enables a seamless agent-creation flow



Related Work

- Utilized RAG and REALM frameworks to ground the model's responses in domain-specific documents
- The system incorporates MemGPT to manage long-term conversation history and overcome fixed context window limits
- Referenced Character.AI and Inworld as benchmarks for no-code avatar creation platforms

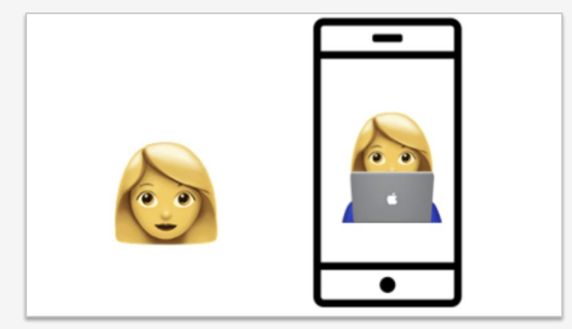
Use Case

Mock Interview Practice

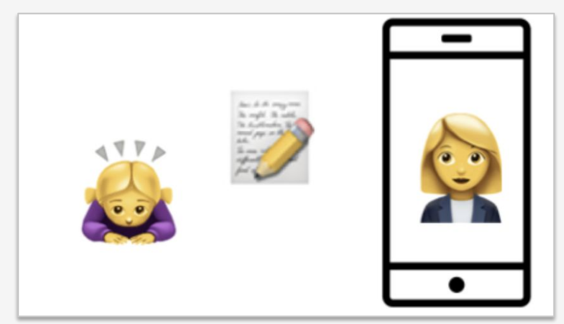
Job seeker initiates the preparation process

User inputs company and role details

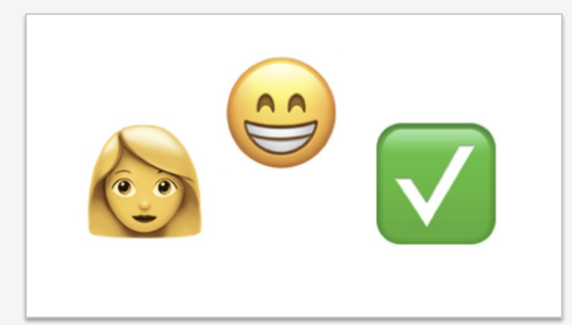
AI generates custom interviewer and scenario



User engages in the mock interview

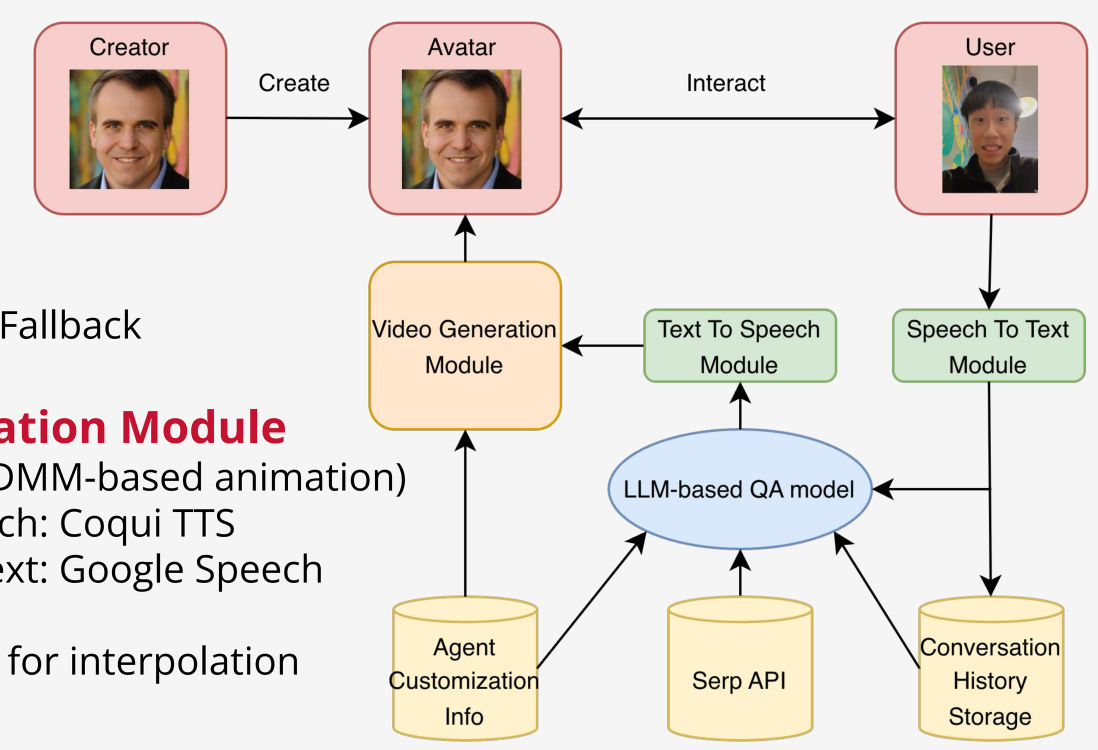


AI offers instant feedback and improvements



User feels confident and fully prepared

Dual-Retrieval RAG Pipeline



Serp API

- Web Search Fallback

Video Generation Module

- Sadtalker (3DMM-based animation)
- Text to Speech: Coqui TTS
- Speech to Text: Google Speech Recognition
- Google FILM for interpolation

Knowledge Base Retrieval

- PDF extraction + chunking
- Vector store indexing

Conversation Memory Retrieval

- Embedding-based semantic search (Sentence-BERT)
- FAISS for efficient nearest neighbor

Experimental Design

Dataset

- Utilized the UDA-QA dataset, comprising PDFs from diverse domains: Wikipedia, Academic, and Finance, paired with corresponding Q&A sets

Pipeline

- Text extraction and chunking from source PDFs
- Vectorizing and indexing documents using a FAISS vector database
- Retrieving the Top-K relevant chunks per query (where K=1,5,10,50)
- Generating grounded answers using Gemini 2.0/2.5 Flash

Goal

- Evaluate the impact of retrieval depth (Top-K) on the quality and groundedness of RAG-generated responses

Results

Performance Across Top-K Values Retrieved (100 questions):

Wikipedia	Top-1	Top-5	Top-10	Top-50
F1 Score	0.35	0.44	0.48	**0.56**
BLEU Score	0.14	0.19	0.19	**0.26**
ROUGE-L	0.33	0.41	0.43	**0.50**
BERTScore	0.88	0.89	0.89	**0.91**

Academic	Top-1	Top-5	Top-10	Top-50
F1 Score	**0.09**	0.08	0.07	0.08
BLEU Score	**0.02**	0.00	0.01	**0.02**
ROUGE-L	**0.09**	0.06	0.07	**0.09**
BERTScore	**0.83**	0.82	0.81	0.80

Finance	Top-1	Top-5	Top-10	Top-50
F1 Score	**0.11**	0.07	0.06	0.07
BLEU Score	**0.04**	0.01	0.01	0.02
ROUGE-L	**0.10**	0.07	0.07	0.08
BERTScore	**0.82**	0.80	0.79	0.79

Evaluation Metrics

- F1, BLEU, and ROUGE-L evaluate lexical (exact word) overlap
- BERTScore evaluates semantic (meaning) consistency

Discussion & Key Findings

Domain-Specific Sensitivity

- Broader retrieval context significantly boosts performance for general knowledge queries like Wikipedia
- Specialized domains (Finance/Academic) degrade at high depths due to "context noise," emphasizing the need for high precision (Top-1) over broad recall

Optimal Retrieval Strategy

- Increasing retrieval quantity is not always beneficial due to the observed precision-noise trade-off
- A Top-3 retrieval depth proved optimal to balance context with noise reduction, further supplemented by conversation history and web search

Generative Nature of RAG

- High BERTScore (>0.80) despite low exact-match metrics confirm the system prioritizes semantic correctness
- The model effectively synthesizes and paraphrases information rather than merely copying retrieved text (verbatim reproduction)

Future Work

- Reduce avatar generation latency by implementing GPU batching
- Migrate the database architecture to Pinecone, PostgreSQL, and Redis
- Conduct systematic benchmarking against established baselines to further validate system robustness

Conclusion

CyberLife AI's no-code platform leverages dual-retrieval RAG and persistent memory to achieve high accuracy (BERTScore > 0.8). Users can build domain-specific agents simply by uploading documents, eliminating the need for programming expertise.