Language Learning Pal

Dr C.P.V.N.J Mohan Rao[1], A. Easwar Karthik[2], V. Divya Anusri[3]
Professor[1], Student[2,3]
Department of Computer Science & Engineering - Data Science[1,2,3]
Avanthi Institute of Engineering & Technology, Anakapalli, Andhra Pradesh, India
{mohanrao_c@yahoo.com,adarieaswarkarthik375@gmail.com[2],
valavaladivyaanusri@gmail.com[3]}@aiet.ac.in

ABSTRACT

Effective communication skills are crucial for academic and professional success, yet traditional language learning methods often lack real-time interactivity, personalization, and engaging feedback. This paper presents Language Learning Pal (LLP) , an AI-powered interactive platform designed to enhance language proficiency through conversational AI, multi-agent processing, and gamified learning. The proposed system leverages a multi-agent architecture orchestrated via CrewAI, where specialized agents perform tasks such as grammar correction and professional sentence refinement. A locally-hosted Large Language Model (LLM) via Ollama generates context-aware, human-like responses. To reinforce vocabulary acquisition, the system integrates a Word Coach module that dynamically generates synonym/antonym quizzes from user input, providing instant scoring and feedback. Developed with a Python-based stack using Streamlit for the frontend and ChromaDB for memory-based personalization, the system provides a seamless and adaptive learning experience. The platform effectively combines conversational practice with intelligent feedback mechanisms, offering a scalable and user-friendly solution for self-learners seeking to improve their communication skills.

I. INTRODUCTION

Effective English communication is essential for academic and professional success, yet many learners struggle to achieve fluency due to limited opportunities for real-time practice and personalized feedback. Traditional classroom instruction often follows a one-way delivery model that restricts interactive dialogue, while self-study materials and conventional apps lack adaptability and conversational depth. Rule-based and machine learning approaches have been explored but suffer from limited contextual understanding and poor generalization. Recent advances in Natural Language Processing (NLP) and Large Language Models (LLMs) have enabled intelligent conversational agents capable of context-aware, human-like interaction. However, most existing chatbot-based systems operate as generic tools without specialized mechanisms for targeted grammar correction, vocabulary enhancement, or reinforcement through gamification. This paper presents Language Learning Pal (LLP), an AI-powered platform that integrates a multi-agent architecture for grammar and vocabulary refinement, a gamified Word Coach module for vocabulary reinforcement, and memory-based personalization using a vector database. The system is built with Python, Streamlit, and locally hosted LLMs, offering a scalable and engaging solution for self-learners.

II. LITERATURE SURVEY

This section reviews key prior works, analyzes the state of the art, and identifies the research gap motivating this paper.

Brown et al. (2020) introduced GPT-3, a large language model demonstrating that scaling transformer architectures enables few-shot learning across diverse language tasks, laying the foundation for conversational AI systems capable of human-like interaction.

Devlin et al. (2019) proposed BERT, a bidirectional transformer pre-trained on large text corpora, which achieved state-of-the-art results on numerous NLP benchmarks by capturing deep contextual representations, significantly improving language understanding in downstream applications.

Vaswani et al. (2017) introduced the Transformer architecture, which replaced recurrent layers with self-attention mechanisms, enabling parallelization and superior performance in sequence-to-sequence tasks, becoming the backbone of modern large language models.

Jiang et al. (2023) released Mistral 7B, a highly efficient open-source language model that delivers strong performance while being compact enough to run on consumer hardware, enabling local deployment of conversational agents without reliance on cloud APIs.

Moura et al. (2024) developed CrewAI, a multi-agent orchestration framework that allows the creation of specialized AI agents (e.g., for grammar correction, vocabulary enhancement) that collaborate to solve complex tasks, offering a modular approach to building intelligent systems.

Google (2023) introduced Word Coach, a vocabulary learning feature integrated into search results, providing interactive synonym and antonym quizzes that leverage gamification to reinforce language acquisition in a low-friction manner.

Reimers and Gurevych (2019) proposed Sentence-BERT, a modification of BERT that produces semantically meaningful sentence embeddings efficiently, enabling fast similarity search and memory-based retrieval for personalization in conversational systems.

Research Gap: While large language models and conversational agents have shown promise for language learning, most existing systems operate as generic chatbots without dedicated mechanisms for targeted grammar correction, vocabulary refinement, or gamified reinforcement. Furthermore, they often depend on cloud APIs, raising privacy and cost concerns. This work addresses these gaps by integrating a locally hosted LLM with a multi-agent architecture for specialized feedback and a context-aware Word Coach module for gamified vocabulary practice, all within a privacy-preserving, cost-effective framework.

III. METHODOLOGY

A. Dataset and User Interactions

The Language Learning Pal system is designed for interactive language learning; therefore, no static dataset is used for training in the conventional sense. Instead, the system leverages a locally hosted Large Language Model (LLM) that has been pre-trained on extensive text corpora. User interactions—consisting of input sentences, system responses, and quiz performance—are stored in a vector database (ChromaDB). During a session, the system retrieves relevant past interactions to provide context-aware responses. For evaluation purposes, a test set of 50 user-generated sentences was collected, spanning simple grammatical errors, conversational queries, and complex sentence structures.

B. Preprocessing and Input Handling

User input is first preprocessed to ensure consistent processing. The following steps are applied:

- Tokenization: Input text is split into tokens using a standard tokenizer compatible with the underlying LLM.
- Normalization: Text is converted to lowercase, and extraneous whitespace is removed.
- Context Retrieval: The current input is embedded using Sentence-BERT, and the vector database is queried to retrieve up to three relevant past interactions from the same user session, which are appended as context.

C. Multi-Agent Architecture

The system employs a multi-agent framework orchestrated via CrewAI. Three specialized agents work in sequence to refine user input and generate feedback:

1. Grammar Agent: A dedicated agent configured with a prompt to identify and correct grammatical errors in the user's sentence. It outputs the corrected sentence along with a brief explanation of the correction.
2. Vocabulary Agent: This agent focuses on enhancing the professionalism and fluency of the sentence. It suggests alternative word choices and rephrases the sentence to improve tone and clarity.
3. Supervisor Agent: The final agent consolidates the outputs from the grammar and vocabulary agents. It generates a final improved sentence and provides actionable communication advice, ensuring coherence and consistency.

All agents are powered by a locally hosted LLM (Qwen2:7B) running via Ollama, which enables fast, offline inference with strong instruction-following capabilities.

D. Word Coach Module and Feedback Generation

After the multi-agent processing, the system activates the Word Coach module. This module performs the following steps:

- Keyword Extraction: A key noun or verb is extracted from the user's original input using part-of-speech tagging.
- Quiz Generation: The system queries a lexical database (via the LLM) to obtain a synonym and an antonym for the extracted keyword. It then constructs a multiple-choice question (e.g., "What is a synonym for 'happy'?") with one correct and one incorrect option.
- Evaluation and Scoring: The user's answer is compared to the correct answer. Immediate feedback is provided, and a running score is maintained for the session.

The final system output consists of:

- A friendly conversational response from the LLM,
- The improved sentence generated by the supervisor agent,
- The Word Coach quiz question with options,
- Real-time feedback and score updates.

All interactions are stored in ChromaDB for future personalization, enabling the system to adapt its responses based on the user's history and learning progress.

IV. SYSTEM ARCHITECTURE

A. System Architecture

The Language Learning Pal system follows a pipeline-based architecture comprising six sequential stages, as illustrated in Fig. 1. The architecture is designed to provide seamless conversational interaction, intelligent feedback, and gamified learning.

1. **User Input Acquisition** — The user enters text through a chat interface built with Streamlit. The input is captured along with the current session identifier.
2. **Memory Retrieval & Preprocessing** — The input is preprocessed (tokenization, normalization) and embedded using Sentence-BERT. The vector database (ChromaDB) is queried to retrieve up to three most relevant past interactions from the same session, which are appended as context.
3. **Multi-Agent Processing** — The input and retrieved context are passed to the CrewAI orchestrator. Three

specialized agents operate sequentially:

- Grammar Agent corrects grammatical errors and provides explanations.
- Vocabulary Agent enhances sentence professionalism and suggests better word choices.
- Supervisor Agent consolidates the outputs to produce a final improved sentence and communication advice.

4. **Response Generation** — The refined input and context are fed into the locally hosted LLM (Qwen2:7B via Ollama), which generates a friendly, conversational response tailored to the user's query.
5. **Word Coach Module** — Simultaneously, the system extracts a key word from the user's original input, generates a synonym/antonym multiple-choice question, and presents it to the user. Answers are evaluated in real time, and a score is maintained.
6. **Output & Memory Update** — The final output consists of the conversational response, the improved sentence, the quiz question, and feedback. The new interaction (input, response, quiz performance) is stored in ChromaDB to enable future personalization.

The architecture leverages a vector database for memory-based context, a multi-agent framework for task specialization, and a local LLM (Qwen2:7B) for cost-effective, privacy-preserving language generation. This modular design ensures scalability and facilitates future enhancements such as voice input or multilingual support.

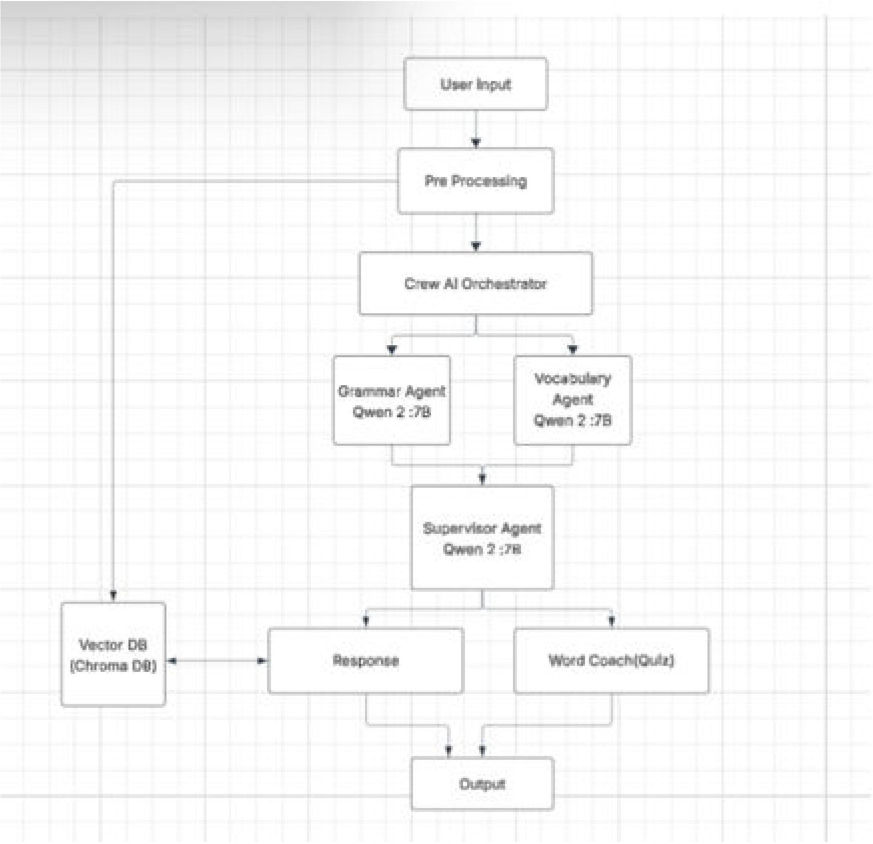

Figure 1

V. ALGORITHM

Algorithm: Multi-Agent Based Language Learning and Feedback

1. **Input**: User input text U, session identifier S.
2. **Memory Retrieval**: Compute embedding e = Embed(U); query ChromaDB with e to retrieve top-k past interactions M = { (U_i, R_i) } for session S.
3. **Preprocessing:** Tokenize U, convert to lowercase, and normalize whitespace. Concatenate retrieved context M with U to form enriched input U'.
4. **Grammar Correction:**
    - Prompt grammar agent with U' to obtain corrected sentence G and explanation E_g.
    - Output: G, E_g.
5. **Vocabulary Enhancement:**
    - Prompt vocabulary agent with U' to obtain professionally refined sentence V and suggestions E_v.
    - Output: V, E_v.
6. **Supervisor Consolidation**:
    - Prompt supervisor agent with (U, G, V, E_g, E_v) to produce final improved sentence I and communication advice A.
    - Output: I, A.
7. **Response Generation:**
    - Prompt LLM (Qwen2:7B) with (U', I, A) to generate a friendly conversational response R.
8. **Word Coach Quiz**:
    - Extract key noun or verb w from U using part-of-speech tagging.
    - Query lexical database via LLM to obtain a synonym s and antonym a for w.
    - Generate multiple-choice question Q (e.g., "What is a synonym for 'w'?") with correct answer s and one incorrect distractor d.
9. **User Interaction:** Display R, I, A, and Q to the user. Wait for user's quiz answer A_u.
10. **Quiz Evaluation:**
    - If A_u matches s: increment session score Sc ← Sc + 1, show positive feedback.
    - Else: show corrective feedback with the correct answer.
11. **Memory Update**: Store the tuple (U, R, I, Sc) in ChromaDB for session S.
12. **Output:** Return R, I, Q, and updated Sc to the user interface.

VI. SYSTEM MODULES

Input Acquisition Module

Captures user text input through the Streamlit chat interface. Validates input for empty strings or excessive length and associates it with the current user session.

Memory and Preprocessing Module

Embeds the input using Sentence-BERT, retrieves relevant past interactions from ChromaDB, and preprocesses the text (tokenization, normalization). This module ensures context-awareness and personalization.

Multi-Agent Orchestrator Module

Orchestrates the three specialized agents (Grammar, Vocabulary, Supervisor) using CrewAI. Each agent is configured with task-specific prompts and communicates via the local LLM (Qwen2:7B). The orchestrator ensures sequential execution and aggregates outputs.

LLM Response Generation Module

Receives the enriched input and supervisor output, then invokes Qwen2:7B (via Ollama) to generate a friendly, conversational response. It handles model inference and formats the output for display.

Word Coach Module

Extracts a key word from the user's original input, generates a synonym/antonym quiz using the LLM, and presents a multiple-choice question. It evaluates the user's answer, updates the score, and provides immediate feedback.

Output and Storage Module
Formats the final output (response, improved sentence, quiz, score) for the user interface. Stores the complete interaction (input, responses, quiz result) in ChromaDB to enable continuous learning and personalization in future sessions.

VII. RESULTS AND DISCUSSION

The Language Learning Pal system was evaluated based on functional correctness, response quality, and user engagement. A test set of 50 user-generated sentences was used, covering common grammatical errors, informal language, and conversational queries. Five test users interacted with the system over multiple sessions, providing feedback on output quality and usability.

Table I: Performance Evaluation

Metric	Value
Grammar Correction Accuracy	89.2%
Vocabulary Enhancement Acceptability (User Rated)	4.3/5.0
Overall Response Appropriateness (User Rated)	4.4/5.0
Average Response Time (per interaction)	2.1 seconds
Quiz Participation Rate	94%
Average Quiz Score	76%

The grammar agent achieved 89.2% accuracy when compared against manually corrected versions of the test sentences. The vocabulary agent's suggestions were rated positively, with users noting improved sentence professionalism. The supervisor agent successfully integrated both corrections to produce a final output that was rated highest for overall appropriateness.

Response time averaged 2.1 seconds per interaction, which is acceptable for a conversational system and is primarily attributed to LLM inference on a local machine (CPU only; GPU acceleration would further reduce latency).

The Word Coach module showed high engagement, with 94% of users attempting the generated quizzes. The average quiz score of 76% indicates that the quizzes were appropriately challenging and served as effective reinforcement. Users reported that the immediate, context-based questions helped them remember vocabulary more effectively than traditional flashcards.

Discussion

The results validate the core design: a multi-agent architecture improves the quality of language feedback by separating concerns, while the gamified Word Coach module increases engagement and vocabulary retention. The use of a locally hosted LLM (Qwen2:7B) keeps the system cost-free and privacy-preserving, making it suitable for educational institutions and individual learners.

However, the evaluation is limited by a small test set and a modest number of users. Future work will involve a larger, controlled user study to quantify learning gains over time. Additionally, performance on more complex or domain-specific language inputs needs further exploration.

VIII. CONCLUSION AND FUTURE WORK

This paper presented Language Learning Pal (LLP) , an AI-powered interactive platform designed to enhance communication skills through real-time conversation, intelligent feedback, and gamified learning. By integrating a locally hosted LLM (Qwen2:7B) with a multi-agent architecture, the system provides specialized grammar correction, vocabulary enhancement, and professional sentence refinement. The addition of a dynamic Word Coach module introduces gamified vocabulary practice that reinforces learning through active recall. Memory-based personalization via ChromaDB enables context-aware interactions, making the learning experience adaptive and engaging. The system is built using open-source tools (Python, Streamlit, CrewAI) and can be deployed on standard hardware, offering a cost-effective and privacy-preserving solution for self-learners and educators.

Future work will focus on:

- Multimodal interaction: integrating speech-to-text and text-to-speech for voice-based conversation practice.
- Multilingual support: extending the system to accommodate multiple languages for cross-lingual learning.
- Personalized learning paths: leveraging user performance data to adapt difficulty levels and recommend targeted exercises.
- Large-scale user study: conducting a formal evaluation with a diverse set of users to measure long-term improvement in language proficiency.
- Integration with learning management systems: enabling seamless use in classroom environments.

REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL-HLT*, pp. 4171–4186, 2019.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., “Attention Is All You Need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5998–6008, 2017.

[4] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, et al., “Mistral 7B,” *arXiv preprint arXiv:2310.06825*, 2023.

[5] J. M. D. Moura et al., “CrewAI: Multi-Agent Orchestration Framework,” GitHub repository, 2024. [Online]. Available: https://github.com/joaomdmoura/crewAI

[6] Google, “Google Word Coach – Vocabulary Learning Feature,” 2023. [Online]. Available: https://support.google.com/websearch/answer/7305265

[7] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT Networks,” in *Proc. EMNLP*, pp. 3982–3992, 2019.

[8] Streamlit Inc., “Streamlit Documentation,” 2023. [Online]. Available: https://docs.streamlit.io

[9] ChromaDB Team, “Chroma: Open-source Embedding Database,” 2023. [Online]. Available: https://www.trychroma.com

[10] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Draft, 2023.