Vietnamese – German University
DEPARTMENT OF COMPUTER SCIENCE

Frankfurt University of Applied Sciences
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Music Theory Chatbot

Author: Nguyen Minh Thuan, Ho Duy Phuc
Matriculation Number: 10421057 - 10421048
Supervisor: Dr. Dinh Quang Vinh

PROJECT REPORT

Binh Duong, Vietnam
SoundSage Repository

Contents

Chapter 1

Abstract

This project focuses on developing a custom chatbot, "SoundSage" using LangChain and Chainlit, designed to provide assistance for music theory, piano playing and music production. Drawing insights from four books, the chatbot helps users understand music theory and gives advices for music production. Using LangChain, the system integrates custom datasets derived from these books to offer tailored advice. The chatbot's user-friendly interface, developed in Chainlit, allows for intuitive interaction, where users receive step-by-step explanation for music theory. By synthesizing key concepts from music theory and production frameworks, "SoundSage" serves as a comprehensive tool for learning and creativity in the musical domain. This project demonstrates how AI and natural language processing can be applied to music education and production, offering users an accessible way to deepen their understanding of music theory and enhance their production skills in their creative journeys.

Keywords: *LangChain, Chainlit, custom chatbot, natural language processing*

Chapter 2

Introduction

2.1 Project Overview

The integration of artificial intelligence into educational tools has revolutionized the way users engage with specialized knowledge domains. This project focuses on the development of a chatbot designed to assist users in exploring music theory, music production, and piano techniques. By leveraging advanced AI frameworks and technologies, the chatbot combines natural language understanding with domain-specific expertise to deliver accurate and context-aware responses.

The chatbot was implemented using LangChain, a versatile framework for building applications powered by large language models (LLMs). LangChain facilitated the integration of a Retrieval-Augmented Generation (RAG) approach, enabling seamless interaction between the language model and a structured knowledge base of music-related content. ChromaDB was employed for storing vectorized data, allowing efficient and accurate retrieval of relevant information. The user interface was developed using Chainlit, ensuring a user-friendly and interactive experience. To handle conversations, the ChatGPT API was integrated, providing robust natural language capabilities. Additionally, a database was implemented to store chat histories, enabling future features such as session continuity and personalized user experiences.

By combining these technologies within the LangChain framework, the chatbot bridges the gap between theoretical knowledge and practical application, offering learners a dynamic and personalized tool for skill development. This report provides an overview of the objectives, implementation methodology, and the potential of this project to support and enhance learning in the field of music.

2.2 Importance of Music Education and Guidance

Music education and guidance are essential for fostering sustainable learning and skill development by providing individuals with accessible and personalized support. By focusing on foundational concepts and interactive, engaging content, the music chatbot minimizes learning barriers and decision fatigue, enabling users to explore music theory and production at their own pace. Through consistent and tailored interaction, users can enhance their knowledge, build confidence, and experience the joy of musical growth. Ultimately, the chatbot empowers individuals to achieve their creative goals by offering structured, reliable, and dynamic resources for learning and exploration.

Chapter 3

Literature Review

3.1 Overview of *Basic Music Theory*

Basic Music Theory by Jonathan Harnum is a foundational guide for understanding and applying key concepts in music theory. The book introduces fundamental elements such as scales, intervals, chords, rhythm, and musical notation, offering a clear and engaging approach to learning. Harnum organizes the content to build progressively, starting with the basics of reading music and advancing to more complex topics like key signatures, chord progressions, and rhythm patterns. Practical exercises and study guides are included throughout to help learners connect theoretical knowledge to real-world musical application. By emphasizing accessibility and enjoyment, the book caters to musicians of all levels, making music theory approachable and relevant. Harnum's conversational style and emphasis on practical learning provide a solid framework for anyone looking to deepen their understanding of music.

3.2 Key Concepts from *Music Theory: From Beginner to Expert*

In *Music Theory: From Beginner to Expert*, Nicolas Carter presents a comprehensive framework for understanding and mastering the fundamentals of music. The book emphasizes the interconnectedness of core elements—harmony, melody, and rhythm—while providing detailed explanations of scales, chords, intervals, and modes. Carter introduces the concept of building musical knowledge progressively, from basic concepts like notes and pitch to advanced topics such as diatonic and chromatic scales, chord inversions, and extended harmonies. A standout feature is the focus on practical application, offering exercises to develop both theoretical understanding and aural skills. The book highlights the importance of the circle of fifths for mastering key signatures and chord relationships, helping learners navigate musical composition and improvisation. By emphasizing the universality of music theory across instruments and genres, Carter encourages readers to integrate theory into their personal musical journeys, fostering both technical skill and creative expression.

3.3 Insights from *Piano for Dummies*

Piano for Dummies by Adam Perlmutter offers an accessible and practical approach to learning the piano, emphasizing the importance of foundational techniques and incremental progress. The book introduces key skills such as proper posture, hand positioning, and basic note-reading while guiding readers through scales, chords, and rhythm exercises. Perlmutter highlights the significance of starting with manageable, beginner-friendly pieces and gradually advancing to more complex musical concepts. By breaking down learning into small, achievable steps, the book minimizes the intimidation often associated with mastering a musical instrument. Additionally, it provides tips for effective practice habits, encouraging consistency and building confidence over time. With its step-by-step structure and engaging tone, Piano for Dummies demonstrates how simple, repeatable actions can lead to significant musical growth, fostering both technical proficiency and enjoyment in playing the piano.

3.4 Perspectives from *The Art of Music Production*

The Art of Music Production by Richard James Burgess provides a comprehensive perspective on the creative and technical processes involved in producing music. The book explores the diverse responsibilities of a music producer, including collaborating with artists, managing recording sessions, and crafting the overall sound of a project. Burgess underscores the importance of blending technical proficiency with interpersonal and artistic skills, showcasing how producers shape the artistic direction while ensuring technical precision. The book also examines the evolution of music production, highlighting how technological advancements have transformed workflows and methodologies. By focusing on adaptability and the producer's ability to unify artistic elements, Burgess presents a detailed overview of the challenges and intricacies of the profession. With its holistic coverage, The Art of Music Production offers valuable perspectives on the role of producers in transforming creative ideas into compelling musical works.

Chapter 4

Technologies Used

In this chapter, we delve into the primary technologies utilized in the development of the chatbot project. These technologies each contribute unique functionalities and advantages that shape the capabilities of the chatbot, from data handling and user interface to complex language processing.

4.1 LangChain

LangChain serves as the foundational framework for the chatbot, providing robust tools for building and structuring conversational workflows. It is instrumental in enabling efficient data retrieval, integration, and management of diverse information sources. LangChain facilitates seamless interactions between various components, allowing the chatbot to process and respond to user inquiries with speed and accuracy. By leveraging LangChain, the chatbot is capable of handling complex conversational contexts and delivering quick, contextually relevant responses, significantly enhancing the overall user experience.

4.2 Chainlit

Chainlit is utilized to design an intuitive, user-friendly interface for the chatbot. Its streamlined setup and specialized features for conversational AI make it an ideal choice for building engaging and interactive user interfaces. Chainlit supports customizable components, enabling users to interact directly with the chatbot in a dynamic and accessible way. The platform's focus on conversational workflows simplifies the UI design process, allowing the team to concentrate on optimizing the chatbot's core functionalities and enhancing user satisfaction.

4.3 ChromaDB

ChromaDB is employed as the primary vector database for the chatbot, enabling efficient storage, retrieval, and management of high-dimensional embeddings. This database is essential for implementing semantic search and contextual understanding, as it allows the chatbot to match user queries with relevant information based on meaning rather than exact wording. By leveraging ChromaDB's performance-optimized architecture, the chatbot achieves rapid and accurate retrieval of contextually relevant data, supporting advanced features such as memory recall and multi-turn conversations. This capability significantly enhances the chatbot's ability to deliver precise and meaningful responses, ensuring a more engaging and informed user experience.

4.4 ChatGPT API

The OpenAI API key serves as a secure authentication mechanism that grants access to OpenAI's language model services. In this project, the API key is essential for two key functionalities: generating conversational responses and embedding data. For conversational capabilities, the API key enables the integration of the ChatGPT model, which processes user inputs and generates accurate, context-aware replies. For embeddings, the API key allows the project to convert textual data into vectorized representations, facilitating efficient storage and retrieval of information using ChromaDB. These combined functionalities are critical for enabling dynamic interactions and ensuring the chatbot delivers relevant, high-quality responses in the domain of music learning.

Chapter 5

System Architecture and Development Process

5.1 Document Retrieval

The `retriever.py` file is responsible for loading documents, splitting them into chunks, and creating a vector store for efficient retrieval.
Vector embeddings are then stored in `chroma.sqlite3` file.

5.1.1 Path Setting:

Defines paths for data and Chroma storage

```
DATA_PATH = "data/"
CHROMA_PATH = "chroma_data/"
```

5.1.2 Load Environments Variables and Documents

```
dotenv.load_dotenv()

documents = []
for file in os.listdir(DATA_PATH):
    if file.endswith('.pdf'):
        pdf_path = os.path.join(DATA_PATH, file)
        loader = PyPDFLoader(pdf_path)
        documents.extend(loader.load())
```

- Environment variables loaded from a `.env` file.
- PDF files are loaded from `DATA_PATH` and appended to the `documents` list.

5.1.3 Documents Splitting:

The loaded documents are split into smaller chunks using CharacterTextSplitter.

```
    text_splitter = CharacterTextSplitter(chunk_size=1000, chunk_overlap=10)
chunked_documents = text_splitter.split_documents(documents)
```

5.1.4 Chroma Client Initialization:

```
    client = chromadb.Client()
    if client.list_collections():
        consent_collection = client.create_collection("consent_collection")
    else:
        print("Collection already exists")
    vectordb = Chroma.from_documents(
        documents=chunked_documents,
        embedding=OpenAIEmbeddings(),
        persist_directory=CHROMA_PATH,
    )
    vectordb.persist()
```

The code sets up a ChromaDB vector database to store and manage document embeddings for tasks like semantic search or retrieval-augmented generation (RAG). It creates a collection, embeds documents using a model like OpenAI's embeddings, and persists the database for efficient similarity-based querying.

5.2 Database Management

The db_utils.py file provides a robust and efficient way to manage conversation data using SQLite.

It ensures that the database is properly initialized and offers a simple interface for saving and retrieving conversations.

Conversation history is stored in conversations.db file.

5.2.1 Database Initialization:

```
    def _init_db(self):
    with self._get_db() as (conn, cur):
        cur.execute('''
            CREATE TABLE IF NOT EXISTS conversations (
                conversation_id TEXT PRIMARY KEY,
                messages TEXT,
                last_updated TIMESTAMP
            )
        ''')
```

The method creates a table named `conversations` if it does not already exist.
The table has three separate columns containing conversation ID, messages and timestamp of last update.

5.2.2 Conversations Saving:

```
    def save_conversation(self, conversation_id: str, messages: list):
    with self._get_db() as (conn, cur):
        cur.execute(
            '''INSERT OR REPLACE INTO conversations
            (conversation_id, messages, last_updated) VALUES (?, ?, ?)
              ''',
            (conversation_id, json.dumps(messages), datetime.now())
        )
        conn.commit()
```

The `save_conversation` method inserts or updates a conversation record in the `conversations` table.
It takes a `conversation_id` and a list of messages as parameters and updates the `last_updated` timestamp to the current time.

5.3 Conversation Handler

The `conversation_handler.py` file is designed to manage the lifecycle of conversations, including generating unique conversation IDs, saving,loading messages and formatting message history.

```
    class ConversationHandler:
```

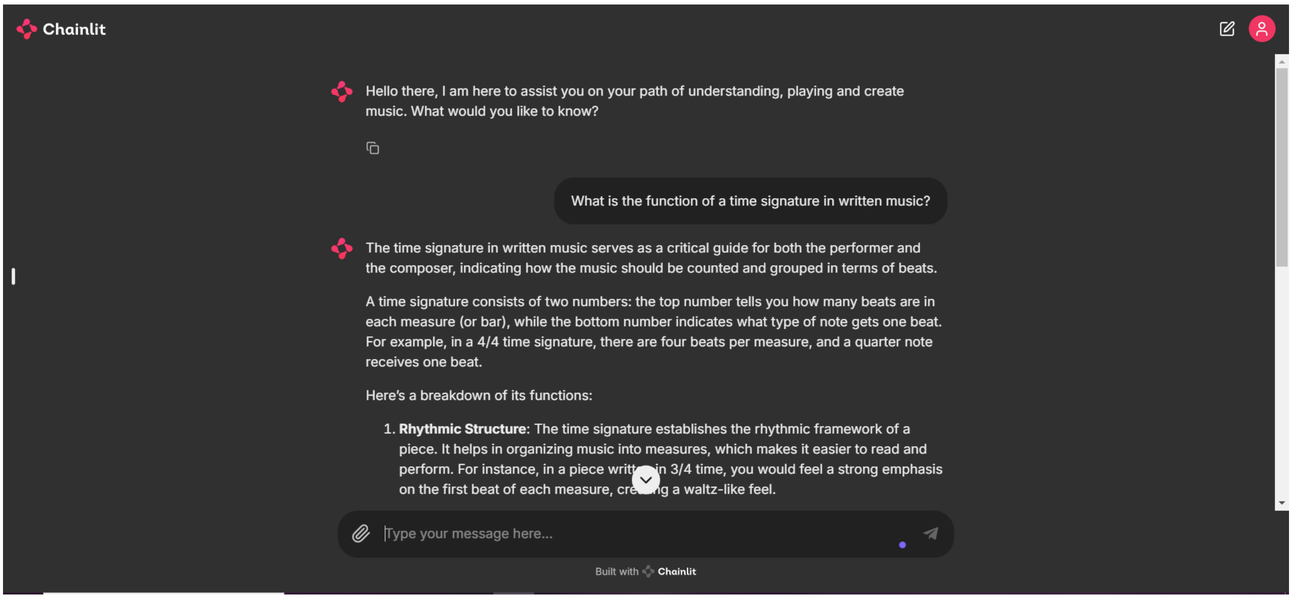
```
    def __init__(self):
        self.db = DatabaseManager()

    def generate_conversation_id(self) -> str:
        """Generate a unique conversation ID"""
        return str(uuid.uuid4())

    async def save_messages(self, conversation_id: str, messages: list):
        self.db.save_conversation(conversation_id, messages)

    async def load_messages(self, conversation_id: str) -> list:
        return self.db.load_conversation(conversation_id)

    async def format_history(self, message_history: list) -> str:
        formatted = []
        for msg in message_history:
            role = "Human" if msg.get("type") == "human" else "Assistant"
            formatted.append(f"{role}: {msg.get(’content’)}")
        return "\n".join(formatted)
```

5.4 User Interface

The UI of this project is built using **Chainlit**, which provides a user interface for interacting with language models.

The UI is customizable through the configuration file `config.toml` and supports themes, custom CSS, and JavaScript.

The `translations` for various UI elements are stored in the translations directory, supporting multiple languages.



Hello there, I am here to assist you on your path of understanding, playing and create music. What would you like to know?

What is the function of a time signature in written music?

The time signature in written music serves as a critical guide for both the performer and the composer, indicating how the music should be counted and grouped in terms of beats.

A time signature consists of two numbers: the top number tells you how many beats are in each measure (or bar), while the bottom number indicates what type of note gets one beat. For example, in a 4/4 time signature, there are four beats per measure, and a quarter note receives one beat.

Here's a breakdown of its functions:

1. **Rhythmic Structure**: The time signature establishes the rhythmic framework of a piece. It helps in organizing music into measures, which makes it easier to read and perform. For instance, in a piece written in 3/4 time, you would feel a strong emphasis on the first beat of each measure, creating a waltz-like feel.

Figure 5.1: Application Interface

Chapter 6

Conclusion and Future Work

6.1 Key Takeaways

The "SoundSage" chatbot project successfully integrates key concepts from four prominent music-focused books—*Basic Music Theory*, *Music Theory: From Beginner to Expert*, *Piano for Dummies*, and *The Art of Music Production*—into a user-friendly tool powered by LangChain and Chainlit. The key takeaway from this project is that breaking down complex topics into manageable, interactive lessons can foster significant learning and skill development in music. By using LangChain for efficient conversational workflows and Chainlit for a seamless, interactive user interface, we were able to create a chatbot capable of offering tailored, practical guidance to users.

The chatbot allows users to ask questions and receive personalized assistance on music theory, piano techniques, and music production, leveraging the foundational principles and strategies from the books. This project has demonstrated the potential of AI in music education, showing that AI-driven systems can provide real-time, interactive coaching for learners of all levels.

Through this endeavor, we also learned the importance of integrating custom datasets and structuring them effectively to enable accurate and contextually relevant responses. The combination of LangChain and Chainlit has proven to be an ideal solution for this type of project, offering both flexibility and scalability to create an engaging and impactful learning experience.

6.2 Potential Enhancements

While the chatbot has already shown great promise in its current form, there are several potential enhancements that could further improve its functionality and user experience:

- **Integration of Multimedia Content:** Incorporating audio clips, video tutorials, and interactive sheet music can make the learning experience more engaging and immersive. This would allow users to hear and see examples of musical concepts in real time, enhancing their understanding.

- **Adaptive Learning Paths:** Implementing AI-driven customization to offer tailored learning paths based on a user's skill level, preferences, and progress can provide a more personalized and efficient learning experience.

- **Gamification:** Introducing features like quizzes, achievements, and progress tracking can increase user motivation and engagement. For instance, users could earn badges for mastering specific topics or completing lessons.
- **Voice Integration:** Enabling voice-based interaction with the chatbot could make the experience even more engaging, especially for users who prefer speaking over typing.
- **Enhanced Contextual Awareness:** By utilizing historical data and combining it with AI advancements, the chatbot could provide deeper, more insightful responses that reflect prior user interactions, enriching the learning experience.

6.3 Challenges and Limitations

Despite the success of the project, there were several challenges and limitations encountered throughout the development process:

- **User Adaptability:** While the chatbot attempts to adapt to user preferences, it may not fully align with the varying skill levels, learning speeds, and specific goals of every individual, leading to a less personalized experience for some users.
- **Potential for Over-Reliance on AI:** Users may become overly dependent on the chatbot for learning, potentially neglecting traditional methods like engaging with instructors, practicing with peers, or using structured educational courses.
- **Limited Interactivity for Practical Application:** While the chatbot provides theoretical guidance, its ability to assist with practical tasks, such as correcting mistakes in music production or piano practice in real-time, is limited. This reduces its utility for hands-on learners.
- **Scalability:** As the knowledge base expands to include more advanced topics or additional resources, maintaining response accuracy and efficiency may become increasingly difficult without significant optimization.