# Educational Song Composition using Large Language Model (LLM) with Synthesized Vocal Singing

Submitted for the Summer Research Internship
as part of the
**Laidlaw Scholars Leadership and Research Program**

by


**Nagyung KIM**

Guide:
**Pierre Dillenbourg**
Professor, EPFL

Supervisor:
**Daniel Carnieto Tozadore**
Scientist, EPFL




**Department of Communication Systems**
**École polytechnique fédérale de Lausanne (EPFL)**

# Abstract

This project fully automates the composition of educational songs, using a Large Language Model (LLM) to generate both lyrics and chord progressions. The system, including synthesized vocals via the DiffSinger model, composes a complete song with vocals in around 50 seconds. It empowers teachers to introduce new concepts through music without requiring any musical background, enhancing classroom engagement. In addition to academic learning, the project supports the development of children's rhythmical and musical skills, making it an effective tool for both education and musical enrichment.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Description

This project fully automates the creation of educational songs using a Large Language Model (LLM), Chat GPT 3.5 turbo. The LLM generates both the lyrics and chord progressions based on user input, and the DiffSinger model synthesizes the vocals. The system enables educators to create complete songs, with vocal synthesis, in about 50 seconds, enhancing the teaching process without the need for musical expertise.

## 1.2 Problem Formulation

The challenge addressed by this project is the need for an engaging, scalable, and automatic way to create educational songs that can help teachers introduce academic concepts, vocabulary, and rhythmical skills to children. Traditionally, teachers lack the time and musical expertise to create such resources. This project overcomes these obstacles by automating the process using AI, allowing songs to be generated quickly and easily.

## 1.3 Motivation

The primary motivation is to create a tool that supports educators in enhancing children's learning experiences through music, without requiring them to have any background in music composition. Previous approaches to integrating music in education often required manual song creation or pre-existing resources, which lacked flexibility and took a significantly longer time to prepare. This made it difficult to implement music dynamically in the classroom. Our automated system, generating songs in real-time, eliminates the dependency on musical knowledge and significantly reduces preparation time, allowing teachers to use music interactively and instantly.

## 1.4 Proposed Solution

The project employs a Large Language Model (LLM) to generate song lyrics and chord progressions, while the DiffSinger model synthesizes the vocals. This approach allows teachers to introduce new concepts through music instantly, supporting academic learning as well as rhythm and musical skill development. The fully automated process reduces complexity and enables teachers to engage students more effectively.

## 1.5 Scope of the Project

The scope of the project is to provide a tool that can generate educational songs across various academic subjects, such as language learning and mathematics, while also fostering musical and rhythmical skills in children. Although the current version of the system is focused on basic educational songs, future improvements could expand its range to more complex musical compositions or subject-specific song generation. Limitations include the reliance on DiffSinger for vocal quality.

# Chapter 2

# System Analysis

## 2.1 Functional Requirements

Functional requirements define what the system must do and its core features:

- The system shall automatically generate song lyrics and chord progressions based on user input.
- The system shall synthesize vocals using the DiffSinger model within 50 seconds.
- The system shall allow teachers to input the topic or mood for the generated song.
- The system shall enable teachers to save or replay previously generated songs for future use.

## 2.2 Non-Functional Requirements

This section details the non-functional requirements, describing how the system will perform its functions.

- The system should generate a complete song with synthesized vocals within 50 seconds.

- The synthesized vocals should have high pronunciation clarity.
- The user interface should be intuitive and easy to use for teachers without technical expertise or musical background.
- The system should handle multiple requests simultaneously without significant delay in response time.
- The system should maintain a high level of stability and not crash during song generation.

## 2.3 Specific Requirements (Hardware and Software Requirements)

This section describes the hardware and software requirements necessary for the project:

### 2.3.1 Hardware Requirements

- Smartphone or any electric device connected to the internet for user interface (for teacher interaction and playback).

### 2.3.2 Software Requirements

- Operating System: Windows 10/11, macOS, or Linux.
- Python 3.8 or later.
- GPT-3.5 API.
- DiffSinger library for vocal synthesis.
- OpenUTAU software for voice synthesis integration.
- User interface framework

## 2.4 Use Case Diagrams and Description

This section includes the use case diagrams and description of the project. Use cases describe how users interact with the system and illustrate the user-system interactions:


Figure 2.1: Use Case Diagram for Song Generation Tool

- **step 1**: The teacher inputs a topic and/or mood into the system.
- **Step 2**: The system generates lyrics and music based on the input.
- **Step 3**: The teacher plays the song in the classroom to engage students.
- **Step 4**: The system saves the generated song for future playback or reuse.

# Chapter 3

# Project design

This project utilizes two models for song generation: **Model A** : Static Song & Dynamic Lyrics and **Model B** : Static Lyrics & Dynamic Song. Each model has its own strengths, weaknesses, potential, and time consumption characteristics, which are detailed below.

## 3.1 Model A: Static Song & Dynamic Lyrics

### 3.1.1 Description

In Model A, the song's melody is predefined, maintaining a consistent musical structure, while the lyrics are dynamically generated to adapt to the melodic constraints. This model operates on the premise that while the musical composition remains invariant, the lyrical content is flexible and generated in response to specific user inputs such as mood, topic, and desired song duration.

### 3.1.2 Operational Flow

The process flow implemented in Model A is meticulously designed to ensure the integration of user inputs into the generation of a harmonized musical output. Below is the

flowchart which delineates the sequence of operations:

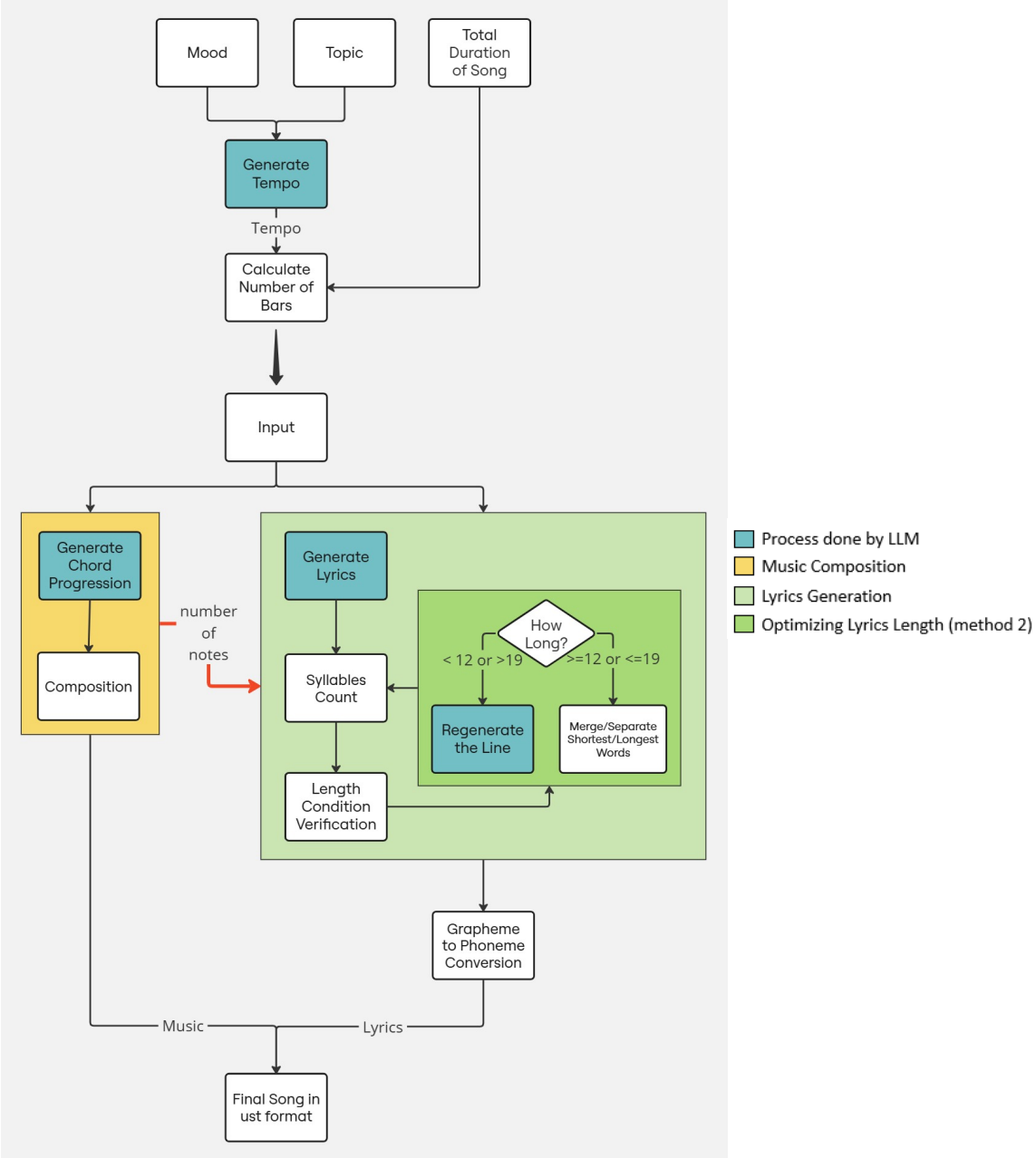


Figure 3.1: Operational Flowchart for Model A: Static Song & Dynamic Lyrics

The sequence of operations is as follows:

- **Tempo Determination**: Initially, the system computes the tempo based on the overall duration of the song as specified by the user, coupled with the mood and thematic input. This tempo setting is critical as it establishes the rhythmic foundation for the subsequent musical structure.
- **Chord Progression Synthesis** (*Performed by LLM*): Subsequent to tempo determination, the LLM synthesizes a chord progression that conforms to the calculated number of musical bars, thereby defining the structural backbone of the composition. The chord progression directly influences the lyrical structure by dictating the syllable count necessary for each segment of the lyrics.
- **Music Composition**: Based on the chord progression provided by the LLM, the program generates the melody. This step involves converting the harmonic structure into a melodic line that complements the generated lyrics. Details of this process will be explained further later.
- **Lyric Generation and Adaptation** (*Performed by LLM*): The LLM generates lyrics that align with the syllabic requirements of the established chord progression. This stage involves a meticulous process of syllable counting and, if necessary, iterative regeneration of lyrical lines to ensure precise alignment with the musical rhythm. Three methods are employed to control the quality of lyrics, which will be explained in detail later.
- **Phonetic Conversion**: As a final step in the lyric preparation, the lyrics are subjected to a grapheme-to-phoneme conversion process. This conversion is crucial for ensuring accurate and clear pronunciation during vocal synthesis. For this conversion, a specialized grapheme-to-phoneme dictionary in YAML format is utilized, which will be further detailed later in this document.
- **Singing Voice Synthesis** (*Performed by OpenUTAU*): Once the music and lyrics are prepared, the information is converted into USTX format, which is then processed by OpenUTAU to synthesize the singing voice. The synthesized vocals are then saved in a WAV file format.

- **Music and Vocal Merging**: In the final step, the harmony and melody, which have been converted from MIDI to WAV, are merged with the synthesized singing voice to create a unified final song file in WAV format.

### 3.1.3 Strengths and Weaknesses

| Strengths | Weaknesses |
|---|---|
| **Performance Adaptability**: Lyrics can be adjusted without altering the underlying music, making it flexible for various educational topics. | **Reduced Lyric Quality**: The repeated regeneration of lyrics to fit a static melody may lead to lower-quality lyrics, especially when trying to match syllable counts. |
| **Consistent Musical Quality**: The melody maintains a high standard and structural integrity, as it is predefined and not altered during the process. | **Lyric Overlap**: There is a risk that lyrics may spill over from one measure to the next, affecting synchronization with the music. |
| **Scalable Composition**: Flexibly adjusts from simple tunes to complex arrangements without losing structural integrity. | **Creativity Limits**: Lyrics are constrained by the fixed musical structure. |

### 3.1.4 Potential

Enhancements in musical quality through advanced techniques such as the incorporation of Markov chains could elevate this model above others. By offering dynamic lyrics alongside enhanced musical creativity, this approach holds the promise of significantly improving both the adaptability and the artistic value of the compositions.

### 3.1.5 Optimizing Lyrics Length: Challenges and Techniques

The following methods are employed to control the quality of lyrics, ensuring they meet the syllable conditions for Model A. Each method has its own set of challenges and

potential benefits:

- **Method 1: Targeted Section Regeneration**
  This method involves reading through the lyrics and identifying lines that exceed or fall short of the required number of syllables. When a line is detected as too long or too short, it is passed back to the LLM for regeneration. The LLM receives instructions specifying whether the renewed lyrics should be shorter or longer to match the correct syllable count.

  For example, if a line is too long, such as:

  – `"Gathering nectar, making honey, oh so lovely" (13 syllables)`

  The system goes through the regeneration process. In this case, the result can be modified to:

  – `"Gathering nectar, honey, sweet and sunny" (11 syllables)`

  However, this regeneration process does not always preserve the original meaning or rhyme. For instance, it might generate a result like:

  – `"Nectar's gift, honey made, so complete" (9 syllables)`

  While this version meets the syllable requirement, it may alter the original tone or meaning. This process is repeated iteratively until the desired syllable count is achieved. The system continues to scan through the lyrics, and whenever another problematic line is detected, the same regeneration process is applied.

  – **Challenges:**
    * More time-consuming as each detection and regeneration may require multiple iterations.
    * Can occasionally disrupt the lyrical flow, causing some sections to sound disjointed or out of context.
  – This method generally provides the best-quality lyrics as it leverages the full flexibility of the LLM.

- **Method 2: Targeted Section Regeneration with Hardcoded Word Manipulation**

  Method 2 is similar to Method 1 in that it reads through the lyrics and identifies lines that are too long or too short. However, instead of always passing the line to the LLM for regeneration, it only does so when the syllable count exceeds or falls short by a significant margin. For minor deviations (within a certain margin), the system hardcodes adjustments by merging or splitting words manually.

  For example, if a line is too long, such as:

  – `"Gathering nectar, making honey, oh so lovely"`

  The system goes through the line and merges the shortest words that can be logically combined. In this case, it would be modified to:

  – `"Gathering nectar, making honey, oh-so lovely"`

  Here, `"oh-so"` is merged and assigned to one note. This approach ensures faster processing when only minor adjustments are needed.

  – **Challenges:**

    * Merging or splitting words may distort pronunciation and disrupt the natural language flow.
    * Requires precise rules for determining which words can be combined or separated without affecting meaning.

  – Generally, this method is more time-efficient compared to Method 1, particularly for lines that are close to the required syllable count.

- **Method 3: Regenerating Entire Lyrics**

  Initially considered, this method involved regenerating the entire set of lyrics if any line failed to meet the syllable requirements. However, due to excessive time consumption and inefficiency, this approach was deemed impractical for operational use and was discontinued.

**Observation & Selection of Method:**

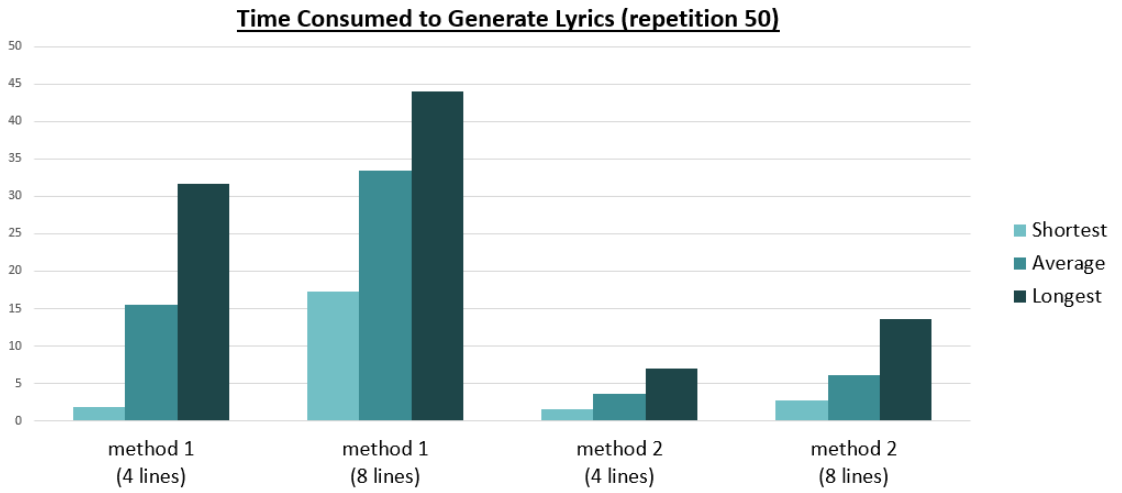


Figure 3.2: Time Consumed to Generate Lyrics (repetition 50), illustrating the efficiency of each method.

- **Method 1** increases time consumption proportionally as the lyric length grows.
- **Method 2** is less affected by lyric length and remains more time-efficient.

For short songs, Method 1 is preferred for quality, while Method 2 is better for longer songs due to its greater time efficiency.

The decision between these methods depends on the specific requirements of the song. For educational songs where clarity and precision in conveying content are crucial, Method 1's thorough approach is advantageous despite its higher time demand. Conversely, for situations requiring rapid production of songs, such as live classroom settings or when creating multiple song versions in a short time, Method 2's efficiency is beneficial.

## 3.2 Model B: Static Lyrics & Dynamic Song

### 3.2.1 Description

In Model B, the lyrics remain static and predefined, ensuring a consistent thematic and lyrical presentation, while the music composition dynamically adapts to these lyrics. This model is particularly useful when the educational content of the lyrics must remain unchanged but requires musical variation to keep the audience engaged and to fit different educational contexts.

### 3.2.2 Operational Flow

The process flow for Model B is designed to dynamically generate music that harmonizes with the fixed lyrics, allowing for real-time musical adaptation based on the thematic elements of the lyrics:

The sequence of operations is as follows:

- **Tempo Determination**: Similar to Model A, the system calculates the tempo based on the overall duration of the song as specified by the user, along with mood and thematic inputs. This step is crucial as it sets the rhythmic foundation for the music, tailored to complement the static lyrics.
- **Lyric Generation** (*Performed by LLM*): Contrary to static lyric integration, in Model B, the LLM generates lyrics dynamically, which fit the provided mood, topic, and tempo. This process ensures that the lyrics are not only thematically appropriate but also rhythmically aligned with the intended musical composition, forming the foundation for subsequent musical synthesis.
- **Chord Progression Synthesis** (*Performed by LLM*): This step involves generating a chord progression that complements the given mood.
- **Music Composition**: Depending on the number of syllables provided by the lyric

generation step, the music is composed to match exactly the syllable count, ensuring that each lyrical segment is properly accentuated by the musical notes.

- **Phonetic Conversion**: This process, identical to that in Model A, involves converting lyrics into phonetic codes to prepare for vocal synthesis, ensuring clarity and accuracy in pronunciation.
- **Singing Voice Synthesis** (*Performed by OpenUTAU*): Like in Model A, the synthesized musical composition and the static lyrics are processed to produce singing vocals, culminating in a harmoniously integrated audio output.
- **Synthesis and Output**: The final step involves synthesizing the complete musical composition, integrating the dynamically generated music with the static lyrics to produce the combined audio output.

### 3.2.3 Strengths and Weaknesses

| Strengths | Weaknesses |
|---|---|
| **Lyrics Quality**: Less restrictive, enhancing lyrical flow and expressiveness. | **Patterned Music**: Music can become predictable, diminishing listener engagement. |
| **Simplified Learning**: One word per note makes it easier for children to learn and sing along. | **Extended Rest Periods**: Long pauses due to lyric scarcity can disrupt the musical continuity. |
| **Harmonical Structure**: Each new word begins at the start of a bar, improving musical harmony and coherence. | **Synchronization Challenges**: Aligning static lyrics with dynamic melodies complicates composition, affecting musical quality. |
| **Controlled Lyric Adaptability**: Precise lyric control benefits educational content by emphasizing key themes or vocabulary. | |

### 3.2.4 Potential

This model holds significant potential for educational settings where the consistency of lyrical content is crucial but can benefit from varied musical accompaniments to enhance learning and engagement.

### 3.2.5 Enhanced Musical Composition Through Syllable Distribution and Duration Manipulation

This section describes the meticulous process of integrating static lyrics with dynamically composed music in Model B, emphasizing the optimization of syllable distribution and note duration to achieve a harmonious and engaging musical output.

#### Syllable Distribution and Duration Manipulation Process

The integration of lyrics into music in Model B employs a sophisticated approach to ensure rhythmic consistency and musical harmony through precise syllable distribution and the manipulation of note durations.

Before delving into the technical process, it is essential to understand the notation used:

- **0**: Quarter rest
- **1**: Quarter note
- **2**: Half note
- **4**: Whole note

The process involves several key steps:

1. **Initial Syllable Count and Analysis**: The lyrics are first analyzed to count the syllables in each line, which are then stored in an array. For the given line `"One`

`plus one is two, it's true"`, the syllable count would be stored as `[1, 1, 1, 1, 1, 1, 1]`.

2. **Brute Force Distribution Algorithm**: The algorithm evaluates all potential ways to distribute these syllables across four bars, aiming for an even distribution to avoid overly sparse or congested measures. It calculates the 'score' for each distribution, selecting the one that minimizes deviations from an average distribution, ensuring rhythmic balance. Here's the core function used to calculate scores:

```
def calculate_score(measures):
    avg = sum(sum(m) for m in measures) / 4.0
    score = sum(abs(sum(m) - avg) for m in measures)
    return score
```

3. **Manipulation of Distribution**: Initial distribution `[1, 1, 0, 0]` is transformed to `[1, 0, 1, 0]` as the music is following 4/4 format. The accent must be placed on the first and third note.

4. **Manipulation of Note Durations**: Depending on the resulting distribution, note durations are adjusted to fill the musical bars more effectively. For instance:

   - Sequences such as `[1, 0, 0, 0]` are consistently transformed to `[2, 0, 0]` to reduce the occurrence of rests.
   - Sequences like `[1, 0, 1, 0]` might be adjusted to `[2, 2]` depending on the tempo, which influences the rhythmic feel of the song.

   Here's how the rhythm transformation is applied based on the tempo:

**Examples**

Consider the lyric line `"One plus one is two, it's true"`, with given mood `happy`:

- Initial Syllable Count Array: `[1, 1, 1, 1, 1, 1, 1]`

- Initial Distribution: `[[1, 0, 0, 0], [1, 1, 0, 0], [1, 1, 0, 0], [1, 1, 0, 0]]`

- Manipulated Distribution: `[[2, 0, 0], [1, 0, 1, 0], [2, 2], [1, 0, 1, 0]]`

The adjustments aim to provide an exciting tone to the music, suitable for a mood of exhilaration with a theme related to learning mathematics. Therefore, in this case, to give cheerful expression to the listeners, the notes with short duration are maintained.

For a different mood and topic, consider the lyrics `"Helping flowers grow strong and tall"` with given mood `sorrowful` :

- Initial Syllable Count Array: `[2, 2, 1, 1, 1, 1]`
- Initial Distribution: `[[2, 0, 0], [2, 0, 0], [1, 1, 0, 0], [1, 1, 0, 0]]`
- Manipulated Distribution: `[[2, 0, 0], [2, 0, 0], [2, 2], [2, 2]]`

In this case, the transformation of `[1, 1, 0, 0]` to `[2, 2]` results in longer note durations, creating a slower and calmer musical tone.

**Impact of Improved Distribution**

The examples illustrate how the syllable distribution and duration manipulation processes adapt to craft music that aligns with both the rhythmic requirements and emotional undertones of educational themes. This nuanced approach prevents music from becoming overly patterned, enhancing engagement. By applying weighted random distributions based on tempo, the composition becomes dynamic and responsive, significantly enriching the musical experience and maintaining the educational impact.

## 3.3 Comparative Analysis of Model A and Model B

The comparative analysis between Model A and Model B involves assessing the efficiency and adaptability of each model in generating lyrics of varying lengths. The key metric for comparison is the time consumed to generate lyrics, as depicted in the graph below:

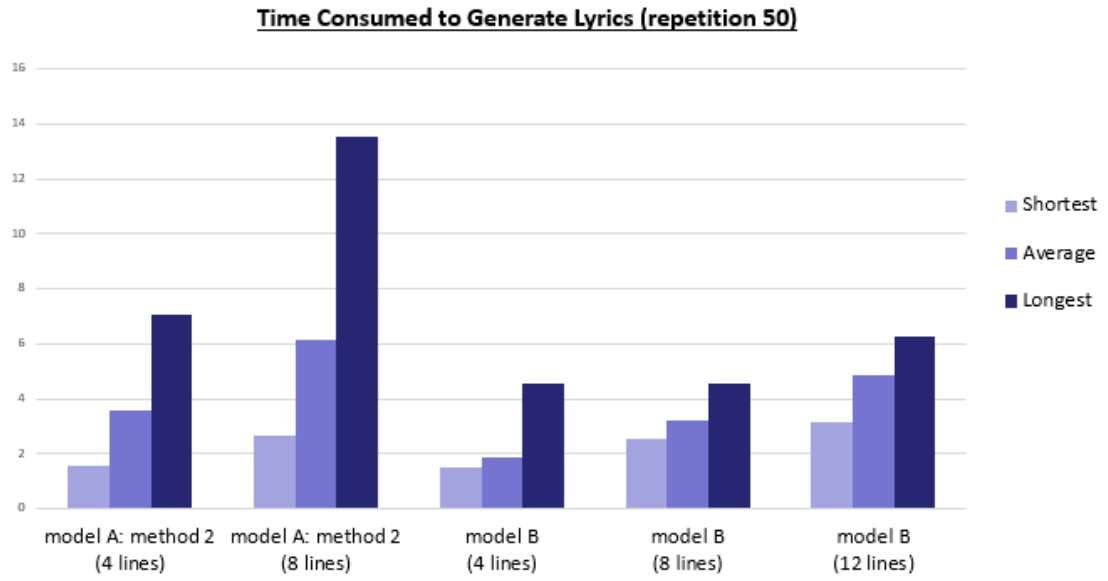


Figure 3.3: Comparison of Time Consumed to Generate Lyrics between Model A and Model B

### Analysis of Time Consumption Across Models

The following points provide a comparative analysis of the time consumed to generate lyrics for Models A and B, illustrating efficiency across varying lyric lengths:

- **Model A (Method 2)**:
  - Shows an increase in time consumption with longer lyrics (8 lines), suggesting scalability challenges as the complexity of the text grows.
  - Performs more efficiently with shorter lyrics, indicating its suitability for brief educational content.

- **Model B**:
  - Demonstrates consistent time consumption across different lyric lengths (4, 8, and 12 lines), highlighting its robustness.
  - Maintains efficiency irrespective of the lyric length, proving advantageous for swiftly generating longer educational content.

This analysis highlights Model B's superior efficiency in scenarios requiring quick content turnaround, pivotal for dynamic educational environments.

**Conclusion**

The analysis demonstrates that Model B currently offers more robust and stable performance in terms of time efficiency, particularly valuable in educational settings where flexibility in lyric length is essential. While Model B excels in operational efficiency, Model A holds potential for producing superior music quality, provided that future advancements or methodologies can significantly enhance its musical output.

## 3.4 Melody and Harmony Music Composition Logic

This section delineates the structured procedures for composing melody and harmony within a predefined musical framework.

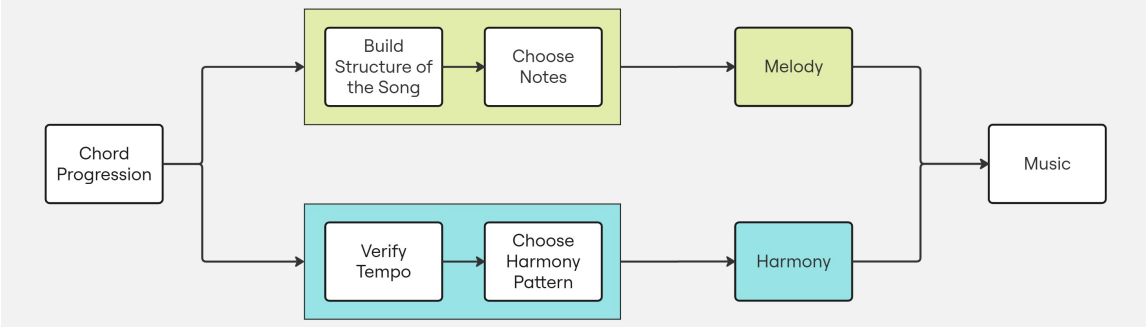


Figure 3.4: Workflow for Music Composition showing the process from chord progression to melody and harmony construction.

### 3.4.1 Melody Composition

Melody composition follows a rigorous process characterized by the following sequential steps:

1. **Chord Progression Analysis**: Initial analysis of the chord progression determines the harmonic foundation.
2. **Melody Structure Development**: Construction of the melody's structure, specifying rest and note patterns to establish rhythmic and melodic contours.
3. **Note Selection**: Selection of notes from the analyzed chords, adhering to constraints designed to maintain musicality:
    - Consecutive notes are limited to a maximum interval of half an octave.
    - Repetition of the same note more than twice consecutively is prohibited.

### 3.4.2 Harmony Pattern Selection

The selection of harmony patterns is dynamically influenced by the tempo of the composition. The system utilizes predefined patterns based on standard piano accompaniment techniques, tailored for educational music. The process follows these steps:

1. **Tempo Verification**: The tempo of the composition is assessed to determine the appropriate weight distribution for harmony pattern selection.
2. **Pattern Groups**: Harmony patterns are grouped into three categories:
    - **Harmony Patterns** (Patterns 1, 5): Focus on enhancing the harmonic richness of the composition.
    - **Simple Patterns** (Patterns 2, 4): Provide straightforward harmonic support, ensuring that the melody remains the focal point.
    - **Arpeggio Patterns** (Patterns 3, 6, 7, 8): Offer rhythmic and textural complexity by arpeggiating the chord.

3. **Weight Distribution**: The system adjusts the weight distribution of pattern selection based on the tempo:

    - **Slow Tempos** ($\leq$ 80 BPM): Weight favors harmony patterns (1, 5) to support a lyrical and expressive accompaniment. These patterns have higher weights due to their ability to add depth to slower pieces. Example: weights [3, 7, 3, 7, 3, 0, 0, 0].
    - **Moderate Tempos** (81-100 BPM): Balanced distribution between harmony and simple patterns, allowing rhythmic support and harmonic texture to coexist. Example: weights [3, 1, 3, 1, 2, 3, 3, 3].
    - **Fast Tempos** ($>$ 100 BPM): Prioritize simple rhythmic patterns to drive forward momentum, with less focus on complex harmonies or arpeggios. Example: weights [1, 1, 1, 1, 1, 1, 1, 1].

4. **Pattern Selection**: Every eight measures, the system re-evaluates the pattern. The selection excludes the previous pattern group to maintain diversity, preventing repetition of harmonic textures.

This method ensures that the harmonic accompaniment is both appropriate for the tempo and mood of the piece, contributing to the structural integrity and emotional quality of the music.

## 3.5 Enhanced Pronunciation Clarity through Phoneme Conversion

The phoneme conversion process, as depicted in Figure 3.5, showcases the step-by-step transformation from graphemes to phonemes using a dictionary. This method ensures accurate pronunciation, crucial for educational technologies where clarity is imperative.

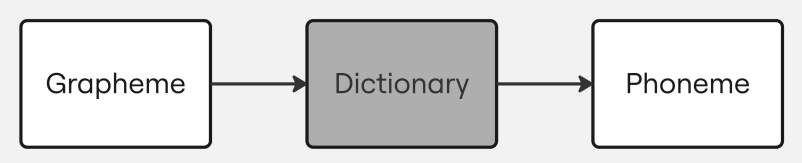


Figure 3.5: Process of converting graphemes to phonemes using a dictionary.

### 3.5.1 Examples of Phoneme Conversion

Below are selected examples demonstrating the conversion of graphemes to their corresponding phonemes:

- The grapheme **act** is converted to phonemes [`@, k, t`].
- The grapheme **activities** results in a more complex set of phonemes [`@, k, t, i, v, i, dd, E, s`], illustrating the nuanced handling of suffix variations.
- The grapheme **address** shows a conversion to [`u, d, r, e, s`], highlighting variations in pronunciation depending on context.
- The grapheme **environmental** shows a conversion to [`e, n, v, I, 3, m, e, n, t, 0, l`], highlighting variations in pronunciation depending on context.

These examples underscore the importance of precise phoneme representation in voice synthesis technologies, particularly in educational contexts where the distinct enunciation of words enhances learning and comprehension.

# Chapter 4

# Results

This chapter presents the results obtained from the implementation of the educational song composition system, focusing primarily on Model B but also exploring the potential of Model A. The discussion highlights the efficiency, adaptability, and educational impact of the system, along with its future potential, as artificial intelligence (AI) models continue to advance.

## 4.1 Efficacy of Model B in Dynamic Song Composition

Model B has demonstrated a high degree of efficiency and adaptability in generating songs based on specific input parameters, such as mood, topic, and tempo. This model operates by first generating lyrics, which are then used to shape the musical composition. Key outcomes of this process include:

### 4.1.1 Time Efficiency

Model B significantly reduces the time required for composing and generating a fully synthesized song, allowing educators to create customized music within a minute. Through

rigorous testing, it was observed that Model B completes the entire song composition process (from lyric generation to vocal synthesis) faster than conventional methods or alternative computational models. The system's ability to generate songs dynamically in real-time presents a valuable tool for educators who need to quickly respond to changing classroom dynamics or student needs.

### 4.1.2 Improved Educational Impact

The rapid song generation enabled by Model B allows educators to incorporate tailored songs that align directly with the educational goals of the lesson. Whether the topic is mathematical concepts or vocabulary enhancement, the generated songs engage students by teaching them through rhythm, melody, and context-specific lyrics. The feedback from early classroom tests has shown that students retain information better and show greater enthusiasm when learning through these dynamically created songs.

### 4.1.3 Scalability and Flexibility

One of the key strengths of Model B is its flexibility in generating a wide variety of songs without compromising on musical coherence. The ability to adapt to various topics and moods makes this model particularly effective for use in different educational settings, from language learning to scientific concepts. Furthermore, the system's scalability ensures that it can handle a high volume of requests for song generation, maintaining its efficiency even in high-demand environments.

## 4.2 Potential of Model A: Quality over Speed

Although Model B excels in efficiency, Model A holds significant potential in delivering higher-quality musical compositions. Unlike Model B, where lyrics guide the music composition, Model A uses predefined melodies, allowing the system to generate lyrics that fit a fixed musical structure. This approach has the following potential advantages:

### 4.2.1 Enhanced Musical Quality

Model A provides a more controlled musical environment by maintaining a consistent, predefined melody. This structured approach ensures that the harmonic and melodic integrity of the music is maintained throughout the composition. For scenarios where musical quality is a higher priority than time efficiency, Model A could deliver more artistically polished outputs.

### 4.2.2 Future Potential with Advancing LLMs

As large language models (LLMs) continue to evolve, we can expect further improvements in both the speed and quality of music generated by systems like Model A. With the potential for more advanced LLMs to handle both complex lyrical phrasing and precise syllabic alignment with musical structures, Model A could eventually surpass Model B in terms of both speed and quality. Furthermore, future iterations of LLMs may reduce the time currently required for regenerating lyrics in Model A, making it a competitive option for both quality and efficiency in real-time song composition.

# Chapter 5

# Conclusions and Future Scope

## 5.1 Conclusion

Both Model A and Model B offer distinct advantages, with Model B excelling in time efficiency and adaptability, while Model A holds the potential for producing more refined musical outputs. The results of this project highlight the immediate benefits of Model B for educators, who can now dynamically create songs in real-time to enhance learning experiences. However, as AI models continue to advance, there is significant potential for improving Model A, enabling it to deliver both high-quality music and fast composition. As these technologies progress, we anticipate a future where AI-driven music composition not only meets but exceeds the demands of educational environments, providing a seamless, engaging, and flexible teaching tool.

## 5.2 Future Work

The successful deployment of Models A and B has opened up various avenues for further research and development to enhance the educational impact of AI-driven song composition. The following initiatives are proposed to expand the capabilities and reach of the current system:

### 5.2.1 Development of a User-Friendly Interface

To make the technology more accessible and engaging for children, a user-friendly interface integrating openUTAU player with a karaoke-like feature is proposed. This interface will allow children to interactively sing along with the songs generated by the system, enhancing both the enjoyment and educational benefit of the music.

### 5.2.2 Integration with Educational Robots

Enhancing interactive learning through technology, the software will be integrated with Alpha-Mini robots. This integration aims to use robots as teaching aids in classrooms, providing a dynamic and engaging way for children to interact with the lessons. The robots will be programmed to perform actions and dances synchronized with the music, thereby reinforcing learning through multi-sensory engagement.

### 5.2.3 Development of a Custom DiffSinger Voicebank

To improve the pronunciation and overall sound quality of the synthesized songs, the development of a custom DiffSinger voicebank is planned. This voicebank will include a wide range of phonemes and intonations, specifically tailored to enhance the clarity and expressiveness of the singing voice, making the songs more appealing and easier to understand for educational purposes.

### 5.2.4 Enhancement of Music Quality with Markov Chains

To further enhance the musical quality of the compositions, research into the use of Markov chains will be conducted. Markov chains will be explored as a method to refine the probabilistic transitions between musical notes, potentially leading to more coherent and pleasing musical sequences. This technique aims to improve the algorithmic composition of music by introducing a level of computational creativity that mimics more closely the nuances found in human-composed music.

These initiatives are designed to build on the current successes of the project by broadening the technological base and enhancing the educational impact of the music generation system. The incorporation of these advancements is expected to not only refine the system's capabilities but also extend its utility and applicability in diverse educational settings.

# Acknowledgements

This study was supported by the Laidlaw Scholars Leadership and Research Program internship and conducted at the Computer-Human Interaction in Learning and Instruction (CHILI) lab, EPFL. I gratefully acknowledge their support.