International Journal of Research in Engineering, Science and Management
Volume 8, Issue 12, December 2025
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing (www.resaim.com)

ArthiX – A Language Agnostic System for Real-Time Text, Speech, and Programming Code Translation

Mahi Paliwal[1*], Kavita Lodhi[2], Poorvi Sharma[3], Sandhya Vishwakarma[4]
[1,2,3]*Student, Department of Computer Science & Engineering (Artificial Intelligence and Machine Learning), Oriental Institute of Science & Technology, Bhopal, India*
[4]*Professor, Department of Computer Science & Engineering (Artificial Intelligence and Machine Learning), Oriental Institute of Science & Technology, Bhopal, India*

***Abstract*: The rapid expansion of global communication and collaborative software development has intensified the need for translation systems that can operate across linguistic and technical boundaries. Existing translators primarily focus on human languages and often fail to address challenges related to voice interaction, offline usage, and programming language understanding. This paper presents ArthiX, a language-agnostic translation system designed to support real-time text and speech translation with automatic language detection, local history storage, and offline capabilities. In addition to natural language translation, the system introduces logical translation of programming code across different languages, followed by real-time compilation to validate correctness. ArthiX is implemented as a desktop application using Python-based Natural Language Processing techniques and speech processing libraries. Experimental evaluation demonstrates that the system provides accurate translations, responsive performance, and improved usability compared to conventional translators. The proposed approach highlights the potential of unified translation platforms in education, software development, and multilingual communication.**

***Keywords*: Language-agnostic translation, Natural Language Processing, Speech translation, Code translation, Real-time compilation, Multilingual systems.**

1. Introduction

Language plays a central role in human interaction, yet it remains a significant barrier in an increasingly globalized digital environment. Students, professionals, and travellers frequently encounter information written or spoken in unfamiliar languages, which limits access to knowledge and effective communication [1]. At the same time, software developers face similar challenges when working with source code written in different programming languages, despite the fact that many programming languages share common logical constructs.

Most existing translation systems focus on either text-based or voice-based human language translation and often require manual language selection. Furthermore, these systems typically rely on continuous internet connectivity and do not provide mechanisms to translate or validate programming code. As a result, users must rely on multiple tools to meet different translation needs [2].

To address these limitations, ArthiX is proposed as a unified language-agnostic translation system. The system is designed to translate text and speech in real time, automatically detect the source language, operate in offline scenarios for selected languages, and store translation history locally. A distinctive feature of ArthiX is its ability to translate programming code logically between languages and compile the translated output instantly, allowing users to verify correctness and execution behaviour [3].

2. Related Work and Background

Early machine translation systems were largely rule-based, relying on manually crafted grammar rules and bilingual dictionaries. While effective for limited domains, such systems struggled with ambiguity and contextual interpretation [2]. The introduction of Statistical Machine Translation improved translation quality by learning probabilistic relationships from bilingual corpora, but these systems still lacked fluency and long-range context handling.

The emergence of Neural Machine Translation marked a major advancement by enabling end-to-end learning of translation tasks using deep neural networks. Encoder–decoder architectures and attention mechanisms significantly improved contextual understanding and translation accuracy. Transformer-based models further enhanced performance by enabling parallel processing and long-distance dependency modelling [4], [5]. They were popularized by the work Attention Is All You Need.

Recent studies have also explored speech-to-speech translation by combining speech recognition and text-to-speech synthesis. However, many existing solutions remain cloud-dependent, lack offline functionality, and do not integrate translation history or programming language support. ArthiX builds upon these research directions by integrating multiple

*Corresponding author: mahi2005paliwal@gmail.com

translation modalities into a single, practical system [1], [7].

3. Problem Statement

Despite significant progress in machine translation, users continue to face several challenges:

- Lack of automatic language detection in many systems.
- Limited support for voice-based interaction and accent variations.
- Dependence on continuous internet connectivity.
- Absence of unified platforms for both human and programming language translation.
- No immediate validation mechanism for translated programming code.

These challenges highlight the need for a comprehensive, language-agnostic system that supports text, speech, and code translation while ensuring usability, accuracy, and reliability [1], [2].

Table 1
Functional requirements of ArthiX

ID	Requirement
FR-1	Accept text input
FR-2	Accept voice input
FR-3	Accept programming code input
FR-4	Auto language or syntax detection
FR-5	Translate content
FR-6	Generate voice output
FR-7	Validate translated code
FR-8	Store translation history

Table 2
Non-Functional requirements of ArthiX

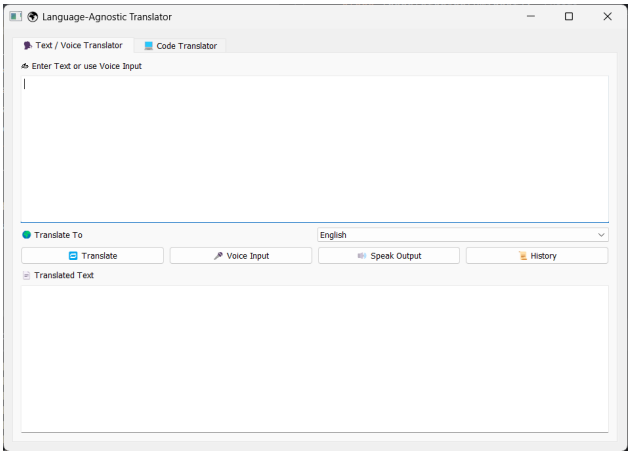
Attribute	**Description**
Performance	Near real-time response
Usability	Simple and intuitive UI
Scalability	Extendable architecture
Reliability	Stable processing
Privacy	Local data storage

4. Proposed System

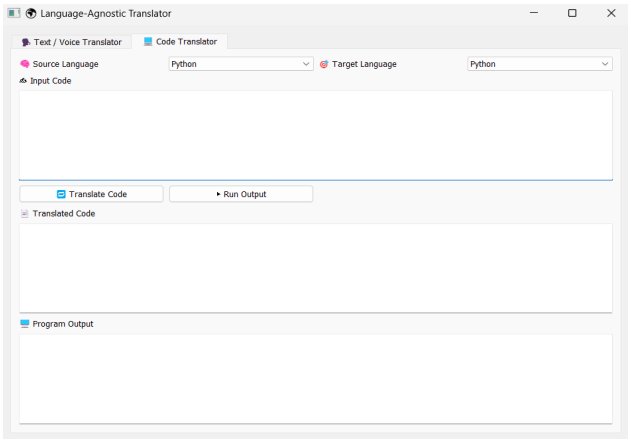
ArthiX is designed as a modular desktop-based translation system that integrates natural language processing, speech processing, and code translation within a single framework. The system architecture consists of three primary layers: the user interface layer, the processing layer, and the data storage layer.

The user interface provides an intuitive environment for entering text, speaking voice input, or submitting programming code. The processing layer handles language detection, translation logic, speech processing, and code compilation. The data storage layer maintains translation history locally to ensure data privacy and persistent access [3].

A key innovation in the proposed system is the real-time code compilation module, which executes or compiles translated code and provides immediate feedback. This feature enables users to validate translated programs and understand execution behaviour without switching tools [7].

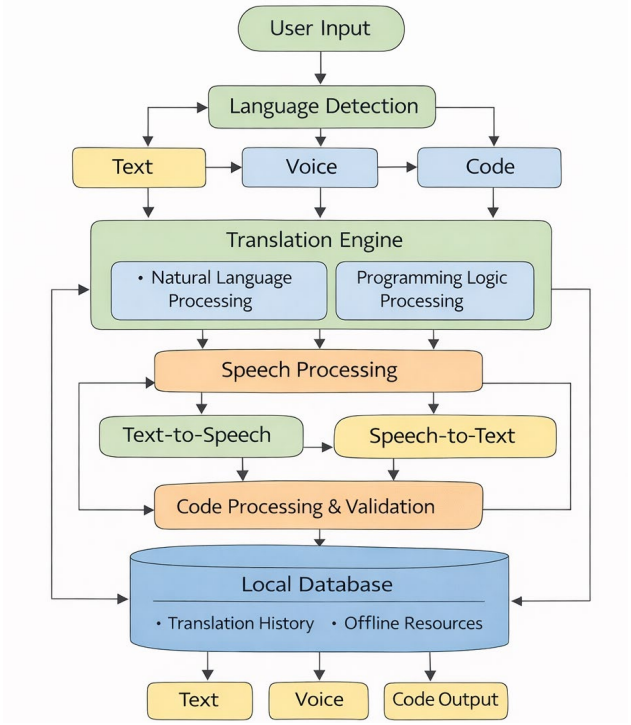


(a)

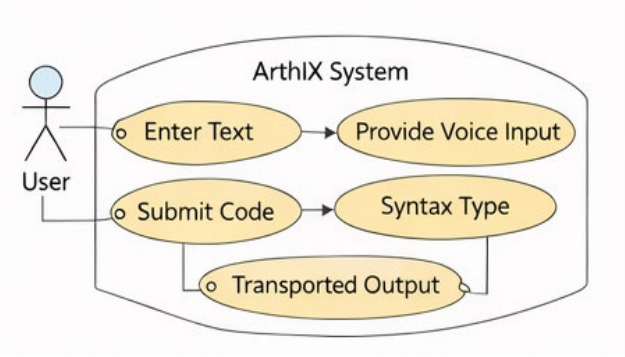


(b)

Fig. 1. ArthiX System Design: (a) Text & Voice Translator (b) Code Translator & Output



(a)



(b)

Fig. 2. System representation: (a) Overall Architecture, (b) Use-case diagram

Table 3
Major modules of ArthiX system

Module	Description
User Interface	Accepts text, voice, and code input
Language Detection	Identifies source language or syntax
Translation Engine	Translates natural language and structured content
Speech Processing	Handles speech-to-text and text-to-speech
Code Processing	Handles logical code transformation and validation
History Management	Stores all translation records
Offline Support	Enables limited translation without internet

5. Methodology

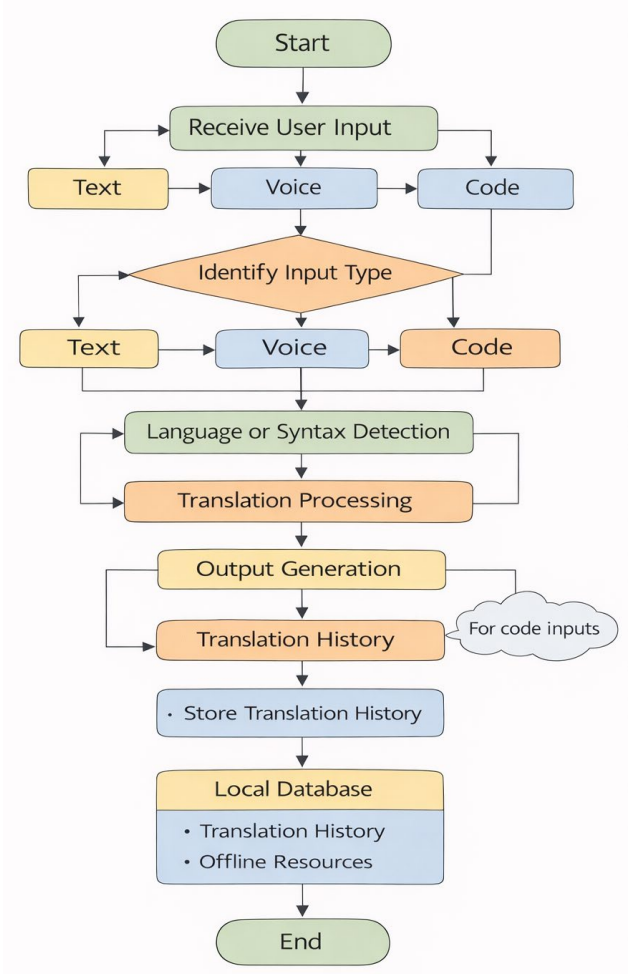


Fig. 3. Workflow of translation process in ArthiX

The development of ArthiX follows an incremental software development approach. Initial development focuses on basic text translation, followed by successive integration of language detection, voice processing, offline support, history storage, and code translation with compilation.

Each module is developed and tested independently before integration, ensuring modularity and ease of maintenance. This approach allows early delivery of functional components and supports continuous improvement throughout the development lifecycle [3].

6. Implementation Details

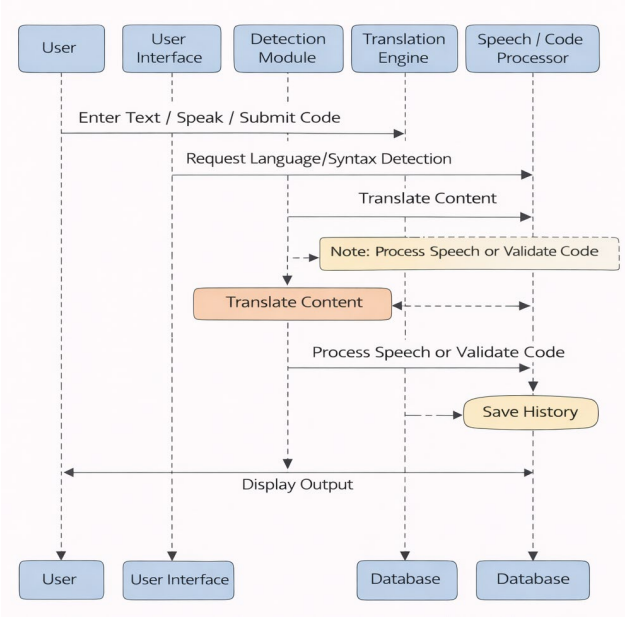
ArthiX is implemented using Python as the core programming language due to its strong ecosystem for NLP, speech processing, and rapid application development. The graphical user interface is developed using PyQt5, providing a responsive and user-friendly desktop experience [8].

Text translation and automatic language detection are performed using NLP-based translation libraries. Speech-to-text functionality enables voice input, while text-to-speech synthesis generates spoken output. Translation history is stored locally using SQLite, ensuring privacy and persistence [9].

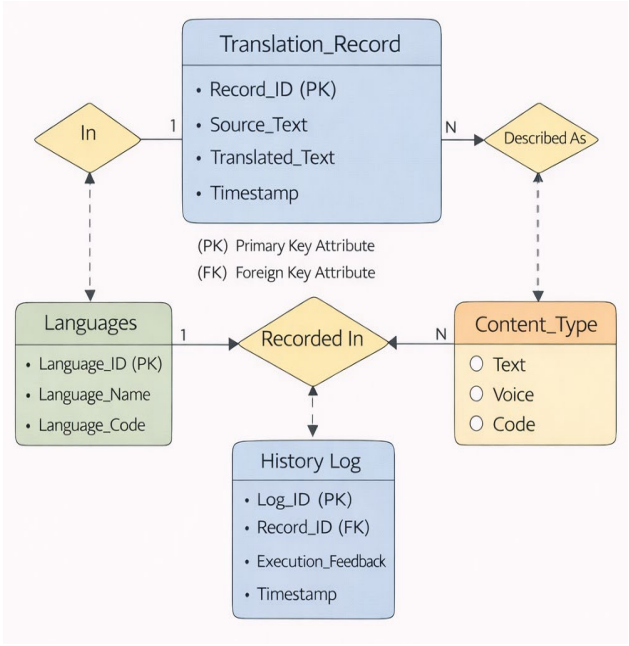
For programming language translation, source code is first converted into an intermediate logical representation. This representation abstracts programming constructs such as loops and conditional statements, allowing equivalent code generation in the target language. The compiled output or execution result is then presented to the user in real time [6].

Table 4
Technologies used in ArthiX

Component	Technology
Frontend	PyQt5
Backend	Python
Language Translation	NLP-based libraries
Speech Input	Speech Recognition
Speech Output	gTTS
Code Handling	Interpreter / Compiler Interfaces
Database	SQLite



(a)



(b)

Fig. 4. Translation and Processing: (a) Sequence diagram, (b) E-R diagram

7. Experimental Results

The system was evaluated using multiple human languages and programming examples. Text translation tests demonstrated accurate and contextually meaningful output with low response time. Voice translation experiments showed reliable speech recognition for clear and moderately accented speech.

The real-time code compilation feature successfully validated translated code in most test cases, providing immediate error feedback when syntax or logical inconsistencies were present. User feedback indicated improved productivity and ease of use compared to traditional translation tools [5], [4].

Table 5
Summary of experimental results

Test Case	Input Type	Outcome
TC-1	Text	Accurate translation
TC-2	Voice	Correct speech processing
TC-3	Code	Correct output generation
TC-4	Offline	Successful execution
TC-5	History	Data stored correctly

8. Applications

ArthiX has broad applicability across various domains:

- *Education:* Assisting students in understanding multilingual study materials and programming concepts.
- *Software Development:* Supporting developers in reading, translating, and validating code written in different languages.
- *Business Communication:* Enabling efficient multilingual communication with international clients.
- *Travel and Accessibility:* Supporting real-time voice translation for travellers and users with reading or speaking difficulties [1], [7].

9. Conclusion and Future Work

This paper presented ArthiX, a language-agnostic translation system that integrates text, speech, and programming code translation within a single platform. By combining automatic language detection, offline functionality, translation history storage, and real-time code compilation, the system addresses key limitations of existing translation tools.

Future work will focus on adding sign language translation as a new feature along with expanding language and programming support, integrating advanced neural translation models for offline usage, and deploying the system on web and mobile platforms. With continued development, ArthiX has the potential to evolve into a comprehensive multilingual communication and development assistant [4], [7].

References

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2023.

[2] P. Koehn, *Statistical Machine Translation*, 1st ed. Cambridge, U.K.: Cambridge University Press, 2010.

[3] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Upper Saddle River, NJ, USA: Pearson Education, 2021.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[5] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learning Representations (ICLR)*, Scottsdale, AZ, USA, 2013.

[7] OpenAI, "Multilingual and cross-lingual models for natural language processing," OpenAI Research, San Francisco, CA, USA, 2023. [Online]. Available: https://openai.com/research

[8] Python Software Foundation, *Python Documentation*, 2024. [Online]. Available: https://docs.python.org/

[9] Google LLC, *Google Cloud Translation API Documentation*, 2024. [Online]. Available: https://cloud.google.com/translate/docs