ADROID Conference Series: Engineering Sciences and Technology
Volume 2, pp.62-72, 2026

AI-Based Intelligent Document Translator And Audio Reader For Telugu Language

K. Sravani[a*], J. Anil[,] V. Bhavika Reddy[,] Muni Sekhar Velpuru

Department of Information Technology, Vardhaman College of Engineering, Hyderabad, India

[a)] sravanikolli63@gmail.com, [b)] jarapulaanil2002@gmail.com, [c)] vbhavikareddy@gmail.com, [d)] munisek@vardhaman.org

Corresponding author: sravanikolli63@gmail.com

Abstract. Accurate machine translation for low-resource languages remains a significant challenge, particularly for technical documents where consistent terminology is essential. This work presents a dictionary-guided post-editing framework for English-to-Telugu document translation. The proposed system operates as a lightweight post-processing layer applied after translation and does not require modifications to the underlying translation model or additional training data. Terminology consistency is achieved through semantic matching with a curated Telugu glossary, ensuring uniform translation of domain-specific terms across the entire document. The system is evaluated using standard translation quality metrics and demonstrates improved performance compared to existing large-scale translation systems. The proposed method achieves a score of 9.23 on a widely used evaluation metric and 42.00 on a character-based evaluation measure, showing consistent improvement in both lexical and morphological accuracy. In addition, a speech synthesis module is integrated to convert translated text into audio form, enhancing accessibility for users. The results indicate that the proposed framework provides a practical and scalable solution for improving translation quality in low-resource language settings, particularly for technical and domain-specific content.

Keywords: Machine Translation; Natural Language Processing; Telugu; Post-Editing; Document Translation; Speech Synthesis.

1. INTRODUCTION

The ability to transfer information across language boundaries has become increasingly critical in domains where equitable access to knowledge is a priority — including education, public governance, healthcare, and technical documentation. Machine Translation (MT) has emerged as a foundational technology in this effort, and recent advances in Neural Machine Translation (NMT) and large language models (LLMs) have produced measurable gains in output fluency and grammatical well-formedness, particularly for language pairs with abundant parallel data. Despite this progress, translation quality for low-resource languages continues to lag considerably, with domain-specific and document-level translation remaining especially difficult — contexts where terminological accuracy and consistency are not merely desirable but operationally necessary [1],[9].

Telugu, one of India's classical languages with over 80 million native speakers, poses distinct challenges for automated translation systems. Its agglutinative morphology, complex inflectional paradigms, and relative scarcity of large-scale, high-quality parallel corpora make it one of the more demanding targets for English–Telugu MT. Contemporary systems addressing this language pair predominantly adopt transformer-based NMT architectures or encoder–decoder frameworks.
[11],[12]. While such models can achieve acceptable sentence-level fluency under favourable conditions, they consistently struggle to preserve the translation of technical terms uniformly across extended documents — a failure mode with direct consequences for practical deployment scenarios such as the translation of educational curricula, engineering manuals, legal instruments, and official administrative records.

This limitation is not confined to smaller or specialised models. Empirical evaluations of state-of-the-art LLMs reveal that, despite strong generalisation across general-purpose tasks, these systems exhibit persistent weaknesses in domain-specific terminology when operating under low-resource conditions [2],[5]. Studies assessing GPT-based architectures specifically report that while surface fluency is often adequate, semantic consistency at the terminology level remains unreliable [2]. Parallel findings in document-level machine translation research further underscore the problem: maintaining lexical

coherence and contextual consistency across sentence boundaries remains an unsolved challenge, even in systems designed with cross-sentence context in mind [4],[6],[7].

Prior work has addressed these limitations through glossary-constrained decoding, domain-specific fine-tuning, document-level context modelling, and automatic post-editing [3],[8],[15]. Automatic post-editing methods have improved translation quality but typically require additional training data or task-specific optimisation, thereby increasing system complexity and deployment costs [3]. Document-level NMT models incorporate contextual embeddings or memory mechanisms to improve coherence [4],[6],[7], but these are computationally expensive and less practical for lightweight deployment in low-resource environments.

Recent work has explored hybrid translation architectures that incorporate LLMs as core components [9],[10]. Although these approaches offer increased adaptability in handling diverse linguistic structures, they do not provide explicit mechanisms for enforcing consistent use of standardised terminology, a requirement that is particularly non-negotiable in technical, scientific, and educational translation contexts. A further limitation of much existing research is its near-exclusive focus on aggregate translation quality metrics, with comparatively little attention given to terminology-level error analysis or the measurement of lexical consistency at the document level [1],[14].

The present work is motivated directly by these gaps. We propose a dictionary-guided post-processing framework for English-to-Telugu translation that diverges from prior approaches relying on glossary-constrained decoding or domain-adapted fine-tuning [4],[6],[7]. Rather than intervening in the translation process itself, the framework operates as a standalone post-editing layer applied to the output of any translation backend. This model-agnostic design enforces the standardised rendering of predefined technical terms through targeted correction, without incurring the computational cost of modifying or retraining the underlying translation model.

Recognising that text-based output alone may be insufficient for users in educational and assistive contexts, the framework additionally incorporates a Telugu text-to-speech (TTS) component. Although multilingual speech and OCR systems have been examined in prior literature [13], the integration of TTS into English–Telugu translation pipelines has received limited attention. Rendering the final translated output as spoken audio meaningfully extends the system's usability beyond traditional text-based interfaces, particularly for low-literacy or visually impaired users.

The primary contributions of this work are summarised as follows:

- A model-agnostic, dictionary-guided post-editing framework for English-to-Telugu translation that enforces terminology consistency across documents without requiring any modification or retraining of the underlying translation model.
- A document-level terminology correction strategy that systematically standardises the translation of domain-specific terms throughout the full document, rather than at the isolated sentence level.
- An integrated Telugu text-to-speech module that converts finalised translated text into audio output, extending system utility to educational and assistive application settings.
- A rigorous empirical evaluation conducted on a publicly available English–Telugu parallel corpus, reporting BLEU and chrF scores and benchmarking performance against multiple LLM-based translation baselines.

The remainder of this paper is organised as follows. Section II surveys relevant literature spanning low-resource machine translation, document-level NMT, and LLM-based translation approaches. Section III details the system architecture and experimental configuration. Section IV defines the methodology and evaluation metrics employed. Section V analyses and interprets the results obtained, and Section VI concludes the paper with a discussion of limitations and directions for future research.

2. Literature Review

A. Machine Translation for Low-Resource Languages

Low-resource machine translation continues to attract significant research attention, driven by persistent obstacles such as the scarcity of parallel training data, morphological richness, and systematic mismatches between general-domain models and specialised content. A comprehensive review by Tafa et al. [11] established that architectural advances in NMT have not closed the

performance gap between high-resource and low-resource language pairs, with technical and formal domains showing the most pronounced deficiencies. Indian languages, including Telugu, are representative of this broader challenge: agglutinative morphology and data sparsity jointly degrade translation quality at both sentence and document granularities.

An mT5-based multilingual translation system for Indian languages proposed by Jha et al. [12] demonstrated improvements in output fluency but yielded limited gains specifically in terminology-level accuracy. Similarly, Asmitha and Kavitha [13] investigated encoder–decoder architectures for English–Telugu translation, observing systematic difficulties in handling compound words and domain-specific expressions. Both studies operate at the sentence level and do not explicitly address the problem of maintaining consistent terminology across full documents — a gap that directly motivates the framework presented in this work.

B. Document-Level Machine Translation

Document-level machine translation (DMT) has emerged as a research direction aimed at overcoming the inherent limitations of sentence-isolated translation by incorporating broader contextual signals. Zhu et al. [4] introduced document-level embeddings to capture cross-sentence dependencies, reporting improvements in output coherence. Zhong et al. [6] and Doan et al. [7] proposed complementary mechanisms, context-aware memory modules and batch-level contextual representations, respectively, to better preserve document-level semantics across translation units.

While these contributions advance contextual modelling in MT, they share a common requirement for architectural modification or additional training procedures, both of which impose computational overhead and reduce practical deployability in low-resource environments. More critically, none of these approaches incorporates mechanisms to enforce standardised translations of domain-specific terms, leaving a functional gap in technical document translation that the proposed framework is designed to fill.

C. Automatic Post-Editing and Terminology Correction

Automatic post-editing (APE) has been investigated as a means of improving translation output quality without altering the underlying translation system. Moon et al. [3] conducted an empirical study of APE applied to NMT outputs, demonstrating measurable improvements in BLEU scores. A fundamental limitation of this approach, however, is its dependence on parallel APE training corpora — resources that are effectively unavailable for low-resource languages such as Telugu, substantially restricting practical applicability.

Karpinska and Iyyer [5] showed that incorporating document-level context does not reliably prevent critical semantic errors in technical translations, suggesting that fluency-oriented models alone are insufficient for terminology-sensitive tasks. Du et al. [15] further argued that purely neural correction strategies are poorly suited to constrained translation scenarios and called for mechanisms that enforce semantic consistency beyond surface-level lexical substitution. Taken together, these findings support the case for hybrid or rule-based post-editing strategies as more appropriate tools for terminology enforcement in low-resource settings — the direction adopted in the present work.

D. Large Language Models for Machine Translation

The application of LLMs as general-purpose translation engines has been evaluated across a range of language pairs and domains. Hendy et al. [2] found that while LLMs achieve competitive performance on high-resource language pairs, quality deteriorates substantially for low-resource languages, with domain-specific terminology posing a persistent challenge. Zhu et al. [9] examined mixture-of-experts LLM fusion strategies for MT. They reported notable instability in terminology usage across generated outputs, a finding consistent with the broader pattern identified in [2]. Hybrid and interactive MT frameworks built around LLMs have also been explored [10], offering greater flexibility in output generation. However, these systems lack robust, built-in mechanisms for constraining terminology to predefined standards in automated large-scale pipelines. The absence of explicit terminology control across LLM-based translation systems constitutes a well-defined gap that the dictionary-guided post-editing layer proposed in this work is specifically designed to address.

E. Semantic Similarity and Embedding-Based Terminology Matching

Semantic vector matching has been widely applied across NLP tasks to identify and align domain-specific terminology across languages. Cagliero and Quatra [8] demonstrated that multilingual

domain-specific word embeddings derived from large document collections can meaningfully improve cross-lingual semantic alignment. However, such embedding induction methods are data-intensive and cannot be readily transferred to low-resource language pairs without substantial supplementary resources. Within MT post-editing specifically, embedding-based methods have been employed to detect semantic drift between source terms and their translated counterparts, enabling targeted correction without full model retraining. The framework proposed in this work extends this line of research by applying semantic vector matching to Telugu technical terminology correction, implemented as a computationally lightweight post-processing layer that operates independently of the translation backend and requires no domain-specific fine-tuning.

F. Accessibility-Oriented Translation Systems

Efforts to broaden the accessibility of multilingual systems have drawn on both OCR and speech-based technologies. Sai Abhishek et al. [14] developed a multilingual OCR system for Telugu text recognition, establishing the viability of automated Telugu text processing in real-world conditions. Latif and Kim [16] examined the applicability of LLMs to clinical text generation with accessibility considerations, identifying a notable absence of speech-based output support for non-English languages as a persistent limitation. Notwithstanding these contributions, the integration of text-to-speech functionality into English–Telugu MT pipelines has received minimal attention. Existing Telugu TTS systems are largely developed and deployed in isolation from translation workflows and do not benefit from upstream terminology-aware corrections. The system proposed in this work addresses this by coupling dictionary-guided post-editing directly with a downstream TTS module, enabling end-to-end translation and audio rendering of technical content for educational and assistive use cases.

3. MATERIALS AND METHODS

The proposed framework is composed of six integrated components, each serving a distinct functional role within the translation and post-editing pipeline:

- **Google Translation API:** Acts as the primary translation backend, receiving English input text and producing an initial Telugu translation. The output of this stage is treated as a draft subject to subsequent terminology correction rather than as a finalised translation.
- **Vector Database:** Stores a curated set of domain-specific English–Telugu terminology pairs, each encoded as a high-dimensional semantic vector. During post-editing, this database is queried to detect and replace technical terms that have been mistranslated, omitted, or rendered inconsistently in the draft translation.
- **Python Processing Pipeline:** Implements the core system logic, encompassing document ingestion, text segmentation, translation orchestration, semantic vector matching, and terminology substitution. Python serves as the integration layer connecting all other components.
- **Flask Web Framework:** Provides the backend application layer responsible for routing HTTP requests between the user-facing interface and the internal translation pipeline, enabling structured communication across system components.
- **Google Text-to-Speech (gTTS):** Accepts the post-edited Telugu text as input and generates a synthesised audio rendering, enabling accessible consumption of translated documents beyond conventional text-based interfaces.
- **Web Interface:** Constitutes the user-facing layer through which source documents are submitted and finalised translation outputs — both text and audio — are retrieved and accessed.

A. System Workflow

The framework executes translation and post-editing through six sequential processing stages, as illustrated in Fig. 1:

1) **Document Upload:** The user submits an English-language source document via the web interface. The document is parsed and decomposed into structured text units suitable for downstream translation processing.

2) **Neural Machine Translation:** The segmented text units are forwarded to the Google Translation API, which produces an initial Telugu translation. This draft output is passed directly to the post-editing stage for correction.

3) **Translation Validation:** The draft translation is systematically scanned to identify terminology-level errors, including instances of untranslated source terms, semantically incorrect domain-specific substitutions, and lexically inconsistent renderings of recurring technical expressions.
4) **Vector Database Lookup:** Terms flagged during validation are matched against the curated terminology database using semantic similarity search over encoded vector representations. Where mismatches are confirmed, erroneous translations are replaced with standardised Telugu equivalents drawn from the database, ensuring uniform terminology usage across the full document.
5) **Final Telugu Text Generation:** The terminology-corrected translation is assembled into a coherent output document and presented to the user as the finalised Telugu text, available for both on-screen review and download.
6) **Text-to-Speech Conversion:** The finalised Telugu text is submitted to the gTTS module, which produces a synthesised audio file corresponding to the translated document. This output supports accessibility requirements and provides pronunciation guidance for users unfamiliar with the written Telugu script.

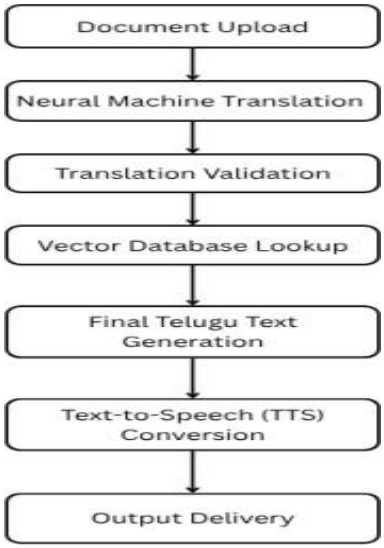

Fig. 1. Flow chart of Document Translator

Fig. 1 depicts the complete end-to-end pipeline of the proposed framework. The process is initiated upon document submission, after which raw text is extracted and forwarded to the Google Translate API for neural translation. The resulting draft output is then subjected to a validation stage in which domain-specific terms are identified and matched against the curated terminology database via semantic vector similarity. Corrections are applied where discrepancies are detected, yielding a terminology-consistent Telugu text document. The pipeline concludes with dual output generation: a finalised translated text and a corresponding synthesised audio file produced by the TTS module.

4. PROPOSED METHODOLOGY

A. System Overview

The framework presented in this work constitutes a complete end-to-end pipeline for English-to-Telugu document translation, incorporating audio rendering as an additional output modality. The central objective of the system is to improve the consistency and reliability of domain-specific terminology in translated technical documents in a low-resource language setting, while deliberately avoiding any reliance on model retraining, architectural modifications, or fine-tuning of the underlying translation backend.

Given a source English document D_{en}, the system produces two outputs: a terminology-corrected Telugu text document D_{te} and a corresponding synthesised Telugu audio file A_{te}. The framework is structured as a post-processing layer that wraps any existing NMT or LLM-based translation system, operating entirely on the translation output rather than intervening in the translation process itself. This design ensures model-agnostic compatibility, allowing the pipeline to be deployed over any translation backend without modification.

B. System Architecture

The pipeline is structured across four sequential and functionally independent stages:

1) Document Preprocessing: The source English document is ingested, parsed, and normalised prior to translation. Text content is extracted from supported input formats, specifically PDF and DOCX, and subsequently segmented into sentence-level units. This granularity of segmentation is adopted to preserve local contextual coherence while maintaining tractable input sizes for the downstream translation component.
2) Baseline Neural Translation: Each sentence unit produced by the preprocessing stage is submitted to a pretrained NMT backend, which generates an initial Telugu translation denoted T_{te}'. This draft translation is not presented as final output; rather, it serves exclusively as the input to the post-editing stage, where terminology-level errors are identified and corrected.
3) Dictionary-Guided Post-Editing: Terminology inconsistencies present in T_{te}' are addressed through a structured post-editing mechanism. Predefined technical terms are located within the draft translation and evaluated against a curated English–Telugu terminology dictionary encoded as a semantic vector database. Where translations are found to be incorrect, inconsistent, or absent, standardised Telugu equivalents are substituted via semantic similarity matching. This matching procedure ensures that corrections are contextually appropriate and that the same technical term receives a uniform translation across all occurrences within the document, enforcing document-level lexical consistency.
4) Text-to-Speech Generation: The terminology-corrected Telugu document D_{te} is submitted to a Telugu TTS engine, which synthesises the corresponding audio output A_{te}. This stage extends the system's utility beyond text-based delivery, enabling accessible consumption of translated content in educational and assistive contexts.

The modular construction of this pipeline is a deliberate architectural choice. By confining terminology correction to a self-contained post-editing layer, the system remains decoupled from the translation backend, allowing quality improvements to be realised through lightweight post-processing alone without any modification to the underlying translation model or its inference procedure.

C. Algorithmic Workflow

The operational logic of the framework is formalised in Algorithm 1, which summarises the sequential execution of the pipeline from source document ingestion through to final output generation. The procedure accepts a raw English document.

D_{en} as input and produces two distinct outputs: a terminology-corrected Telugu text document D_{te} and a synthesised Telugu audio file A_{te}.

1. Input: English document D_{en}
2. Extract and normalise textual content from D_{en}
3. Generate initial Telugu translation T_{te}' via the baseline NMT backend
4. Detect domain-specific and low-confidence term translations within T_{te}'
5. Apply targeted corrections through semantic vector matching against the curated terminology database
6. Reconstruct the corrected translation as the final Telugu document D_{te}
7. Synthesise Telugu audio output Ate from Dte via the TTS module
8. Output: D_{te}, A_{te}

Algorithm 1: Dictionary-Guided Post-Editing Pipeline

The algorithm proceeds in a strictly linear fashion, with each stage conditioning on the output of its predecessor. Terminology detection at line 4 operates over the draft translation T_{te}' produced by the NMT backend, isolating terms that are either domain-specific or flagged as low-confidence renderings. The correction step at line 5 resolves these by retrieving semantically aligned Telugu equivalents from the vector database, applying substitutions that preserve contextual meaning while enforcing uniform terminology across the document. The final two stages assemble the corrected text into D_{te} and derive the audio output A_{te}, completing the pipeline without any interaction with or modification of the underlying translation model.

D. Working Principle

Source English documents are decomposed into sentence-level segments, each submitted independently to a Transformer-based NMT model to produce an initial Telugu translation T_{te}'.

While Transformer-based NMT achieves reasonable general-domain coverage, it exhibits systematic weaknesses in domain-specific terminology, particularly in low-resource settings. The framework addresses this through a confidence-based identification step that flags uncertain term translations in T_{te}'. Flagged terms are matched against the curated English–Telugu terminology database via semantic vector similarity, and incorrect or inconsistent translations are replaced with standardised equivalents. This correction operates at the document level, ensuring uniform terminology across all occurrences within D_{te}.

The corrected document D_{te} is subsequently passed to the speech synthesis module, producing audio output Ate and enabling dual-modality delivery of translated content.

E. Evaluation Metrics

System performance is evaluated using two standard automatic machine translation metrics established in the literature for low-resource and morphologically rich languages.

1) *BLEU Score:* The BLEU (Bilingual Evaluation Understudy) score measures n-gram overlap between system-generated translations and reference translations, computed as:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \, log \, p_n\right) \qquad (1)$$

where BP denotes the brevity penalty applied to penalise translations shorter than the reference, pn represents the modified *n*-gram precision for order *n*, and w_n are uniform weights assigned across *n*-gram orders.

2) *chrF Score:* The chrF metric evaluates character-level *n*-gram precision and recall, making it particularly suited for morphologically rich languages such as Telugu, where word-level metrics inadequately capture morphological variation. The chrF score is defined as:

$$\text{chrF} = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \qquad (2)$$

where β controls the relative weighting of recall over precision, used jointly, BLEU and chrF capture lexical accuracy and morphological adequacy, respectively, providing complementary evaluation of translation quality.

F. Evaluation Results

TABLE 1: COMPARISON OF TRANSLATION QUALITY ACROSS SYSTEMS

System	BLEU	chrF
Gemini-based Translation	2.05	19.76
Grok-based Translation	7.23	37.08
ChatGPT-based Translation	8.79	39.36
Proposed System	**9.23**	**42.00**

Table 1 reports BLEU and chrF scores for the proposed system against three LLM-based translation baselines evaluated on the Samanantar English–Telugu parallel corpus. The proposed system achieves a BLEU score of 9.23 and chrF of 42.00, outperforming all baselines across both metrics. The performance gap is most pronounced against the Gemini-based system, with improvements of 7.18 BLEU points and 22.24 chrF points. Against the strongest baseline (ChatGPT-based translation), the proposed system yields gains of 0.44 BLEU points and 2.64 chrF points, indicating consistent improvement in both

lexical and morphological translation accuracy through dictionary-guided post-editing.

5. RESULTS AND DISCUSSION

A. Quantitative Evaluation

TABLE 2: COMPARISON OF TRANSLATION QUALITY ACROSS SYSTEMS

System	BLEU	chrF
Gemini-based Translation	2.05	19.76
Grok-based Translation	7.23	37.08
ChatGPT-based Translation	8.79	39.36
Proposed System	**9.23**	**42.00**

Table II reports BLEU and chrF scores for the proposed system and three LLM-based baselines evaluated on the Samanantar English–Telugu parallel corpus. The dictionary-guided post-editing framework achieves a BLEU score of 9.23 and a chrF score of 42.00, outperforming all baselines across both metrics.
The Gemini-based system produces the lowest scores (BLEU: 2.05, chrF: 19.76), indicating significant weaknesses in both lexical accuracy and morphological adequacy for English–Telugu translation. The Grok-based system improves over Gemini, achieving a BLEU score of 7.23 and chrF of 37.08, while the ChatGPT-based system yields the strongest baseline performance with a BLEU score of 8.79 and chrF of 39.36.The proposed system outperforms the ChatGPT-based baseline by 0.44 BLEU points and 2.64 chrF points, and exceeds the Gemini-based system by 7.18 BLEU points and 22.24 chrF points. These results demonstrate that dictionary-guided post-editing consistently improves translation quality over direct LLM-based translation across both evaluation metrics.

B. Comparative Analysis

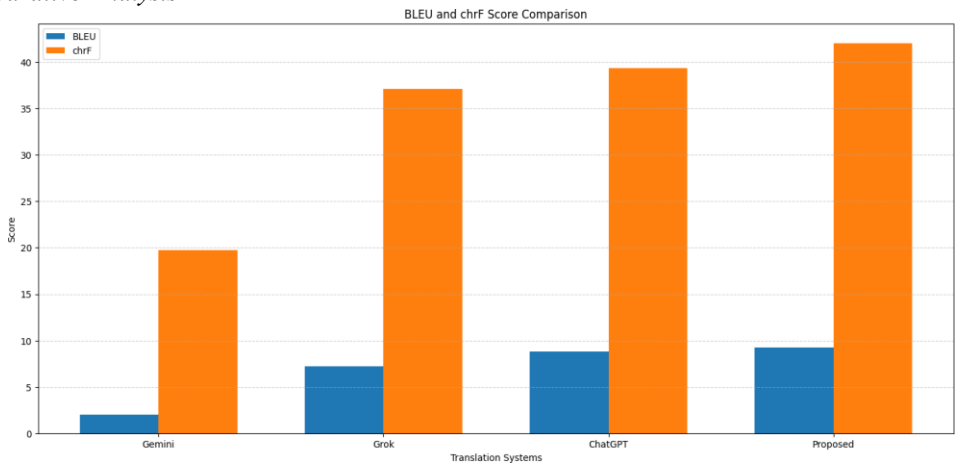

Fig. 2. BLEU and chrF score comparison across translation systems. The proposed system achieves the highest scores on both metrics.

Fig. 2 illustrates the BLEU and chrF score distributions across all evaluated systems. The chrF scores exhibit a more pronounced spread across systems compared to BLEU scores, reflecting the sensitivity of character-level evaluation to morphological variation in Telugu — a characteristic consistent with prior findings on morphologically rich low-resource languages. The narrow BLEU range across baselines and the proposed system (7.23–9.23) contrasts with the comparatively wider chrF spread (37.08–42.00), indicating that word-level n-gram overlap is insufficient to fully capture the morphological improvements introduced by terminology correction. The character-level sensitivity of chrF makes it more responsive to domain-specific term substitutions, rendering it a more discriminative evaluation metric for assessing post-editing effectiveness in morphologically rich, low-resource settings such as English–Telugu.

C. Discussion

The performance gains observed in the proposed system are attributable to the dictionary-guided post-editing mechanism, which enforces standardised Telugu equivalents for domain-specific terms identified during translation validation. LLM-based systems, while producing fluent output, do not incorporate explicit terminology constraints and consequently exhibit inconsistent or incorrect rendering of technical terms, reflected in their lower chrF scores. The BLEU improvement of 2–7 points over LLM baselines is consistent with gains reported in APE literature for low-resource language pairs. The chrF improvement is more substantial, confirming that the post-editing layer addresses morphological-level errors beyond what word-level overlap captures. These results establish dictionary-guided post-editing as an effective and computationally lightweight strategy for improving domain-specific translation quality in low-resource settings without model retraining.

D. System Output

Users interact with the system through a web interface designed for straightforward document submission, accepting digital files.

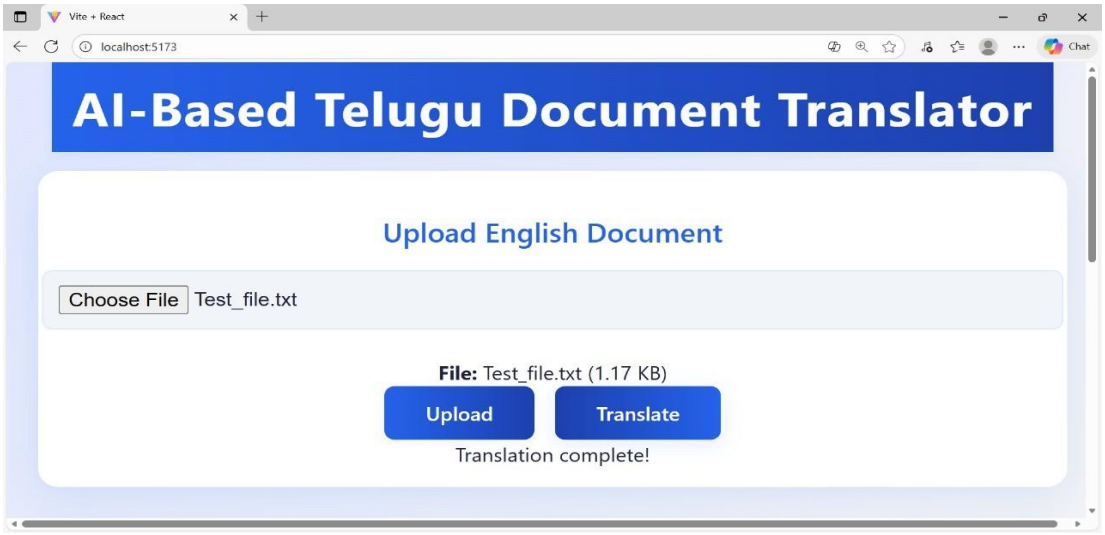

Fig. 3. Document Translator Interface to upload a document

Figure 3 illustrates the system's web interface, through which users may upload a source English document from their local file system and initiate translation by submitting the document via the translate button.

Fig. 4. Response after translating the uploaded document

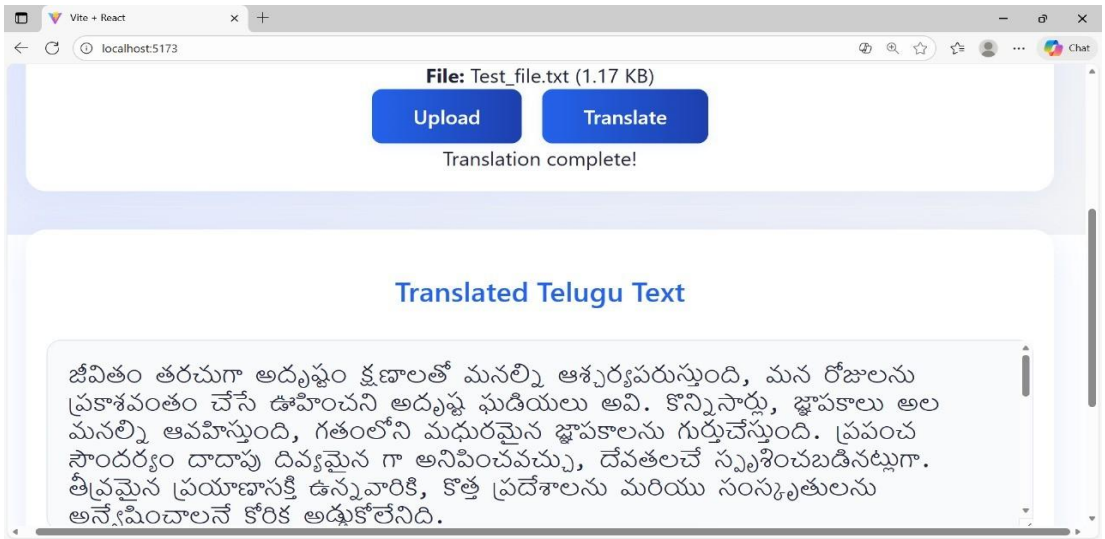

Figure 4 displays the translated output generated for the uploaded document. The interface presents the finalised Telugu text alongside a playable audio rendering, allowing users to read, listen, or engage with both modalities concurrently. This dual-output design particularly benefits users with literacy difficulties or

visual impairments, extending the accessibility of the system beyond conventional text-based translation interfaces.

6. ConCLUSION

This work introduced a dictionary-guided post-editing framework for English-to-Telugu document translation, designed to address terminology inconsistency in technical and domain-specific content. Operating as a model-agnostic post-processing layer over any existing NMT or LLM-based backend, the framework requires no retraining, fine-tuning, or supplementary training data. Semantic vector matching is employed to detect and replace domain-specific terminology errors in the draft translation, enforcing standardised Telugu equivalents uniformly at the document level. An integrated TTS module further extends the pipeline to produce accessible audio output from the corrected text. Evaluation on the Samanantar English–Telugu parallel corpus yields BLEU and chrF scores of 9.23 and 42.00, respectively, surpassing Gemini, Grok, and ChatGPT baselines across both metrics. The magnitude of chrF improvement over LLM baselines confirms that the post-editing mechanism resolves morphological-level translation errors that word-level overlap metrics fail to capture. Collectively, these results establish dictionary-guided post-editing as a practical, computationally lightweight, and scalable strategy for improving domain-specific translation quality in low-resource settings without the overhead of model-level intervention.

A. Future Work

Several directions are identified for extending the proposed framework:

1) **Terminology Database Expansion:** The current vector database covers a constrained set of domain-specific terms. Broadening coverage to encompass additional technical domains, including healthcare, legal, and engineering, is expected to improve post-editing accuracy across a wider range of document types and subject areas.
2) **Cross-Lingual Generalisation: The model-agnostic design** of the framework renders it applicable beyond Telugu. Evaluating its deployment on other morphologically rich low-resource Indian languages, such as Kannada, Malayalam, and Odia, would establish the generalizability of the approach and its broader utility within the Indian language NLP ecosystem.

ABBREVIATIONS

BLEU	Bilingual Evaluation Understudy
chrF	Character n-gram F-score
LLM	Large Language Model
NLP	Natural Language Processing
TTS	Text-to-Speech

CONFLICT OF INTEREST

The authors declare no conflicts of interest regarding the current research.

AUTHOR CONTRIBUTION

K. Sravani: Proposed the research problem and designed the methodology.
J. Anil: Developed the implementation and conducted experiments.
V. Bhavika Reddy: Performed data analysis and validation.
Muni Sekhar Velpuru: Supervised the work and reviewed the manuscript.
All authors discussed the results and contributed to writing the paper.

REFERENCES

[1] T. O. Tafa et al., ”Machine Translation Performance for Low-Resource Languages: A Systematic Literature Review,” in IEEE Access, vol. 13, pp. 72486-72505, 2025.

[2] A. Hendy, M. Abdelrehim, A. Sharaf, V. Raunak, M. Gabr, H. Matsushita, Y. J. Kim, M. Afify, and H. Hassan Awadalla, ”How good are GPT models at machine translation? A comprehensive evaluation,” 2023.
[3] H. Moon, C. Park, S. Eo, J. Seo and H. Lim, ”An Empirical Study on Automatic Post Editing for Neural Machine Translation,” in IEEE Access, vol. 9, pp. 123754-123763, 2021, doi: 10.1109/ACCESS.2021.
[4] L. Zhu et al., ”Document-Level Neural Machine Translation With Document Embeddings,” in IEEE Access, vol. 13, pp. 87015-87025, 2025.
[5] M. Karpinska and M. Iyyer, “Large language models effectively leverage document-level context for literary translation, but critical errors persist,” in Proc. 8th Conf. Mach. Transl., 2023.
[6] K. Zhong, J. Zhang, and W. Guo, “Document-level machine translation with effective batch-level context representation,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jun. 2024.
[7] X. -D. Doan, V. -H. Vu, N. -D. Nguyen and V. -T. Tran, ”Document-Level Machine Translation with Context-Aware Memory,” 2024 16th International Conference on Knowledge and System Engineering (KSE), Kuala Lumpur, Malaysia, 2024.
[8] L. Cagliero and M. L. Quatra, ”Inferring Multilingual Domain-Specific Word Embeddings From Large Document Corpora,” in IEEE Access, vol. 9, pp. 137309-137321, 2021.
[9] S. Zhu, L. Pan, D. Jian, and D. Xiong, “Overcoming language barriers via machine translation with sparse mixture-of-experts fusion of large language models,” Inf. Process. Manage, vol. 62, no. 3, May 2025.
[10] Y. Wang, J. Zhang, T. Shi, D. Deng, Y. Tian and T. Matsumoto, ”Recent Advances in Interactive Machine Translation With Large Language Models,” in IEEE Access, vol. 12, pp. 179353-179382, 2024.
[11] A. Jha, H. Y. Patil, S. K. Jindal and S. M. N. Islam, ”Multilingual Indian Language Neural Machine Translation System Using mT5 Transformer,” 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 2023.
[12] M. Asmitha and C. R. Kavitha, ”Bridging the Language Gap: Enhancing English-to-Telugu Translation using NMT and Encoding Decoding Techniques,” 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024.
[13] B. V. Sai Abhishek, K. Yamuna and T. Anjali, ”Multilingual Translational Optical Character Recognition System for Printed Telugu Text,” 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021.
[14] A. Latif and J. Kim, ”Evaluation and Analysis of Large Language Models for Clinical Text Augmentation and Generation,” in IEEE Access, vol. 12, pp. 48987-48996, 2024.
[15] Y. Du, Y. -F. Ma, Z. Xie and M. Li, ”Beyond Lexical Consistency: Preserving Semantic Consistency for Program Translation,” 2023 IEEE International Conference on Data Mining (ICDM), Shanghai, China, 2023.
[16] Ramesh Saini, ”Enhancing Indian Language Translation Using Machine Learning: A Comprehensive Approach”, 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), pp.1-5, 2023.