AN INTELLIGENT SYSTEM FOR DETECTION, TRANSLATION AND HISTORICAL MAPPING OF ANCIENT INSCRIPTIONS

25-26J-144

Bamunu Arachchige Amadhi Hansani

IT22069436

BSc (Hons) degree in Information Technology Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology Sri Lanka

August 2025

AN INTELLIGENT SYSTEM FOR DETECTION, TRANSLATION AND HISTORICAL MAPPING OF ANCIENT INSCRIPTIONS

25-26J-144

Bamunu Arachchige Amadhi Hansani

IT22069436

BSc (Hons) degree in Information Technology Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology Sri Lanka

August 2025

I. DECLARATION

I declare that this is my own work and this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any other University or institute of higher learning. To the best of my knowledge and belief, it does not contain any material previously published or written by another person except where acknowledgements are made in the text.

Also, I hereby grant to the Sri Lanka Institute of Information Technology (SLIIT) the non-exclusive right to reproduce and distribute my dissertation, in whole or in part, in print, electronic, or other medium. I retain the right to use this content, in whole or in part, in future works (such as articles, publications, or books).

Name	Student ID	Signature
Hansani B.A.A.	IT22069436	

I hereby certify that the above-named candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Name	Signature
Ms. Gaya Thamali Dasanayake (Supervisor)	

II. ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my dissertation supervisor, **Ms. Gaya Thamali Dasanayake**, and co-supervisor, **Mr. Samadhi Rathnayake**, for their invaluable guidance, continuous support, and constructive feedback throughout the course of this research. Their expertise and encouragement have greatly contributed to improving both the quality and depth of this study.

I would also like to extend my appreciation to **Prof. Raj Somadewa**, Department of Archaeology, and **Ms. Sadarasi** for their valuable insights and guidance. Their contributions provided important domain knowledge and additional perspective, which helped strengthen the research direction and ensure the historical and practical relevance of this study.

My sincere thanks go to the academic and administrative staff of the Sri Lanka Institute of Information Technology for their continuous support and for providing the necessary knowledge and resources throughout my academic journey.

I am also grateful to my group members for their collaboration, knowledge sharing, and teamwork during the development of the overall system. Their support and discussions have significantly contributed to the successful completion of this project.

Finally, I would like to thank my family and friends for their unwavering encouragement, patience, and support throughout this research journey. Their motivation has been instrumental in helping me successfully complete this work.

III. ABSTRACT

Ancient inscriptions represent a valuable source of historical and cultural knowledge; however, their interpretation remains a complex and time-consuming task due to the absence of clear word boundaries, degradation of text, and limited availability of linguistic resources. Traditional methods rely heavily on expert knowledge, making the process slow and less accessible for large-scale analysis. This research focuses on the development of a Natural Language Processing (NLP) based solution for word segmentation and translation of ancient inscription text as part of an intelligent inscription analysis system.

The proposed approach addresses the challenge of segmenting continuous character sequences into meaningful words using a probabilistic dynamic programming model. This method leverages word frequency distributions and language modeling techniques to determine the most probable segmentation of input text. Following segmentation, a retrieval-based translation mechanism is employed, utilizing a TF-IDF vector space model combined with cosine similarity to identify the closest matching sentence from a predefined corpus. A confidence-based decision system is integrated to ensure reliability, with a fallback dictionary-based translation applied in low-confidence scenarios.

The system is designed to operate effectively in low-resource environments where large, annotated datasets are unavailable. By combining probabilistic modeling and statistical retrieval techniques, the proposed solution provides an interpretable and efficient alternative to data-intensive neural approaches. Experimental results demonstrate that the system can produce meaningful segmentation and translation outputs, improving the accessibility and usability of ancient inscription data.

Overall, this research contributes toward the digitization and preservation of historical inscriptions by automating key stages of text interpretation, enabling researchers, historians, and the public to better understand and analyze ancient textual content.

Keywords: Natural Language Processing, Word Segmentation, TF-IDF, Cosine Similarity, Ancient Inscriptions

TABLE OF CONTENTS

IV. LIST OF FIGURES

V. LIST OF ABBREVIATIONS

Abbreviation	Full Form
AI	Artificial Intelligence
NLP	Natural Language Processing
OCR	Optical Character Recognition
TF-IDF	Term Frequency - Inverse Document Frequency
DP	Dynamic Programming
ML	Machine Learning
API	Application Programming Interface
JSON	JavaScript Object Notation
UI	User Interface
UX	User Experience
CSV	Comma-Separated Values
UTF	Unicode Transformation Format

Table 1: List Of Abbreviations

1. INTRODUCTION

1.1 Background

Ancient inscriptions represent one of the most significant sources of historical, cultural, and linguistic knowledge. In Sri Lanka, inscriptions written in early scripts such as Brahmi provide valuable insights into ancient civilizations, including social structures, religious practices, and administrative systems. Despite their importance, interpreting these inscriptions remains a challenging and time-consuming task, primarily due to the structural and linguistic complexities associated with ancient scripts.

Unlike modern written languages, ancient inscriptions typically do not contain explicit word boundaries. Text is often engraved as a continuous sequence of characters without spaces, making it difficult to identify individual words. In addition, the physical condition of inscriptions further complicates interpretation. Over time, environmental factors such as erosion, weathering, and physical damage result in partial or distorted characters, increasing ambiguity in reading and understanding the text.

Traditionally, the process of interpreting inscriptions has relied heavily on domain experts such as archaeologists and epigraphists. These experts manually analyze the text, identify possible word boundaries, and derive meaning based on linguistic knowledge and contextual understanding. While this approach ensures accuracy, it is inherently slow, labor-intensive, and not scalable for large collections of inscription data. Furthermore, the limited availability of experts restricts broader access to this valuable historical information.

With the advancement of computational technologies, there is a growing interest in applying Natural Language Processing (NLP) techniques to automate the interpretation of ancient texts. However, conventional NLP methods are primarily designed for modern languages, where large annotated datasets and well-defined linguistic structures are available. In contrast, ancient scripts such as Brahmi are considered low-resource languages, with limited datasets, inconsistent writing patterns, and a lack of standardized grammar. These challenges make it difficult to directly apply data-intensive machine learning or deep learning approaches.

This research addresses these limitations by focusing on the development of a hybrid NLP-based approach for word segmentation and translation of ancient inscription text. The proposed solution combines probabilistic modeling techniques with statistical retrieval methods to effectively process continuous text under low-resource conditions. A probabilistic dynamic programming model is used to segment the input text into meaningful word units by identifying the most likely word boundaries based on frequency distributions. Subsequently, a retrieval-based translation mechanism is employed, utilizing a TF-IDF vector space model and cosine similarity to identify the closest matching sentence from a predefined corpus and generate an appropriate translation.

By integrating these techniques, the system provides an efficient and interpretable approach to automate key stages of inscription analysis. This not only reduces the dependency on manual interpretation but also improves accessibility for researchers, students, and the general public. Ultimately, the proposed solution contributes to the broader objective of preserving and digitizing historical knowledge by enabling more efficient analysis of ancient inscription data.

1.2 Literature Review

The application of Natural Language Processing techniques for text segmentation and translation has been widely studied in the context of modern languages. However, the extension of these techniques to ancient scripts remains a relatively underexplored area due to the unique challenges associated with low-resource and unstructured text data.

Word segmentation is a fundamental task in NLP, particularly in languages that do not use explicit delimiters between words. In languages such as Chinese and Thai, segmentation has been addressed using various approaches, including rule-based methods, statistical models, and machine learning techniques. Early approaches relied on dictionary-based matching, where known words are identified within a continuous text. While effective in controlled scenarios, these methods struggle when dealing with unknown words or variations in spelling.

Statistical approaches introduced probabilistic language models to improve segmentation accuracy. Techniques such as n-gram models and Hidden Markov Models (HMM) have been used to estimate the likelihood of word sequences and determine optimal segmentation paths. More advanced methods utilize dynamic programming algorithms to efficiently compute the most probable segmentation of a given input. These approaches are particularly effective in handling ambiguity, as they consider multiple possible segmentations and select the one with the highest probability.

In recent years, deep learning-based approaches such as Recurrent Neural Networks (RNN) and Transformer models have shown significant success in word segmentation tasks. However, these methods require large annotated datasets for training, which are not available for ancient languages such as Brahmi. As a result, their applicability in low-resource scenarios remains limited.

Translation of ancient text presents an additional challenge. Modern machine translation systems, including Neural Machine Translation (NMT), rely on large parallel corpora to learn mappings between source and target languages. In the absence of such datasets, alternative approaches such as rule-based translation and retrieval-based methods have been explored. Retrieval-based translation systems operate by comparing input text with a predefined corpus and selecting the closest matching example. These systems are

particularly suitable for low-resource environments, as they do not require extensive training data and provide interpretable results.

Several studies have demonstrated the effectiveness of TF-IDF vectorization combined with cosine similarity for measuring textual similarity in information retrieval tasks. By representing text as numerical vectors, these methods enable efficient comparison of sentences based on their structural and lexical features. When applied to translation, this approach allows the system to identify semantically similar sentences and retrieve corresponding translations from a limited dataset.

Despite these advancements, existing research primarily focuses on either segmentation or translation as separate tasks. There is a lack of integrated solutions that address both problems within a unified framework, particularly for ancient inscription text. Additionally, most existing systems are designed for modern languages and do not account for the unique challenges posed by degraded text, inconsistent spelling, and limited linguistic resources.

1.3 Research Gap

Although significant progress has been made in the fields of word segmentation and machine translation, several limitations remain when applying these techniques to ancient inscription text.

One major limitation is the reliance on large annotated datasets in modern NLP approaches. Deep learning-based models, while highly effective, require extensive training data, which is not available for ancient scripts such as Brahmi. This makes it difficult to apply such models in low-resource environments.

Another key gap is the lack of systems capable of handling continuous text without predefined word boundaries. Most existing segmentation techniques are optimized for modern languages with well-defined linguistic structures, and their performance degrades when applied to unstructured ancient text.

Furthermore, existing translation systems are predominantly designed for modern languages and depend on parallel corpora for training. In the context of ancient inscriptions, such datasets are scarce or nonexistent, limiting the effectiveness of traditional translation methods.

Additionally, there is a lack of integrated solutions that combine both word segmentation and translation within a single pipeline. Most approaches address these tasks independently, resulting in fragmented workflows that reduce efficiency and accuracy.

1.4 Research Problem

The interpretation of ancient inscription text presents a unique and complex challenge due to the inherent characteristics of early writing systems. Unlike modern languages, ancient scripts such as Brahmi are typically written as continuous sequences of characters without explicit delimiters to indicate word boundaries. This absence of spacing creates significant ambiguity in identifying meaningful lexical units, making word segmentation a critical yet difficult task. A single sequence of characters may be segmented in multiple valid ways, each producing different meanings, thereby increasing the complexity of accurate interpretation.

In addition to structural ambiguity, the physical condition of inscriptions further complicates the problem. Many inscriptions have been subjected to environmental degradation over long periods, resulting in faded, incomplete, or distorted characters. These imperfections introduce uncertainty into the recognition process, which subsequently affects the reliability of segmentation and translation. Even minor inaccuracies in character recognition can propagate through the system, leading to incorrect word formation and misinterpretation of meaning.

Another major challenge lies in the limited availability of linguistic resources for ancient languages. Modern Natural Language Processing techniques, particularly those based on deep learning, require large-scale annotated datasets and parallel corpora for effective training. However, for ancient scripts such as early Sinhala or Brahmi, such resources are scarce or nonexistent. This lack of data makes it impractical to apply conventional data-driven approaches, thereby necessitating alternative methods that can operate effectively in low-resource environments.

Furthermore, existing solutions in the domain of text processing tend to address word segmentation and translation as separate tasks. This separation results in fragmented workflows where errors in segmentation directly impact the quality of translation. Without a unified framework that integrates both processes, it becomes difficult to ensure consistency and accuracy in the overall interpretation of inscription text.

Another limitation of current approaches is their dependence on exact word matching or rigid linguistic rules, which are not well-suited for ancient scripts characterized by spelling variations and inconsistencies. These variations arise due to differences in writing styles, phonetic representations, and historical evolution of language. As a result, systems that rely solely on exact matching fail to capture underlying patterns within the text, reducing their effectiveness in real-world scenarios.

Given these challenges, there is a clear need for a robust and adaptable solution capable of handling unstructured, low-resource text data. The problem addressed in this research is therefore the development of an efficient method to accurately segment continuous ancient inscription text into meaningful words and generate reliable translations, despite the absence of large datasets, clear linguistic structures, and consistent textual patterns.

1.5 Research Objectives

1.5.1 Main Objective

The main objective of this research is to design and develop a Natural Language Processing-based system capable of performing accurate word segmentation and translation of ancient inscription text, enabling efficient interpretation, analysis, and accessibility of historical information under low-resource conditions.

1.5.2 Sub-Objectives

To achieve the main objective, the research focuses on the following sub-objectives:

- **To analyze the structural and linguistic challenges associated with ancient inscription text**, particularly the absence of explicit word boundaries, presence of noise, and inconsistencies in character representation.
- **To develop a probabilistic word segmentation model** capable of identifying optimal word boundaries within continuous text by leveraging word frequency distributions and statistical language modeling techniques.
- **To implement an efficient segmentation algorithm using dynamic programming**, ensuring that the most probable sequence of words is selected while minimizing computational complexity.
- **To incorporate a confidence-based decision system** that evaluates the reliability of translation outputs and applies fallback strategies, such as dictionary-based translation, in cases of low confidence.
- **To enhance the interpretability and usability of the system**, enabling researchers, historians, and general users to understand and analyze ancient inscription text with reduced reliance on expert intervention.
- **To contribute toward the digitization and preservation of historical knowledge**, by providing an automated and scalable solution for processing and interpreting ancient inscriptions.

2. METHODOLOGY

2.1 Methodology

2.1.1 System Overview

The proposed system is designed as a Natural Language Processing (NLP)-based pipeline to perform word segmentation and translation of ancient inscription text. The primary objective of this component is to transform continuous sequences of characters, obtained from earlier stages such as character recognition, into meaningful and interpretable language.

The overall workflow of the system follows a structured pipeline consisting of multiple interconnected stages. These include preprocessing, word segmentation, validation, translation, and output generation. Each stage is designed to address specific challenges associated with ancient text processing, particularly the absence of word boundaries, limited linguistic resources, and high levels of ambiguity.

The system begins by accepting a sequence of recognized characters as input. This text is typically unstructured and does not contain spaces or clear delimiters. The preprocessing stage standardizes the input by removing noise and normalizing character representations. Following this, the segmentation module applies a probabilistic model to identify the most likely word boundaries within the continuous text. The segmented output is then validated using a predefined dictionary to ensure consistency and reduce errors.

The translation module operates on the segmented text using a retrieval-based approach. Instead of generating translations from scratch, the system compares the input with a predefined corpus of known sentence patterns and retrieves the most similar match using statistical similarity measures. Finally, a confidence-based mechanism evaluates the reliability of the output and applies fallback strategies when necessary.

This pipeline-based design ensures that each stage contributes to improving the overall accuracy and interpretability of the system while maintaining efficiency and scalability.

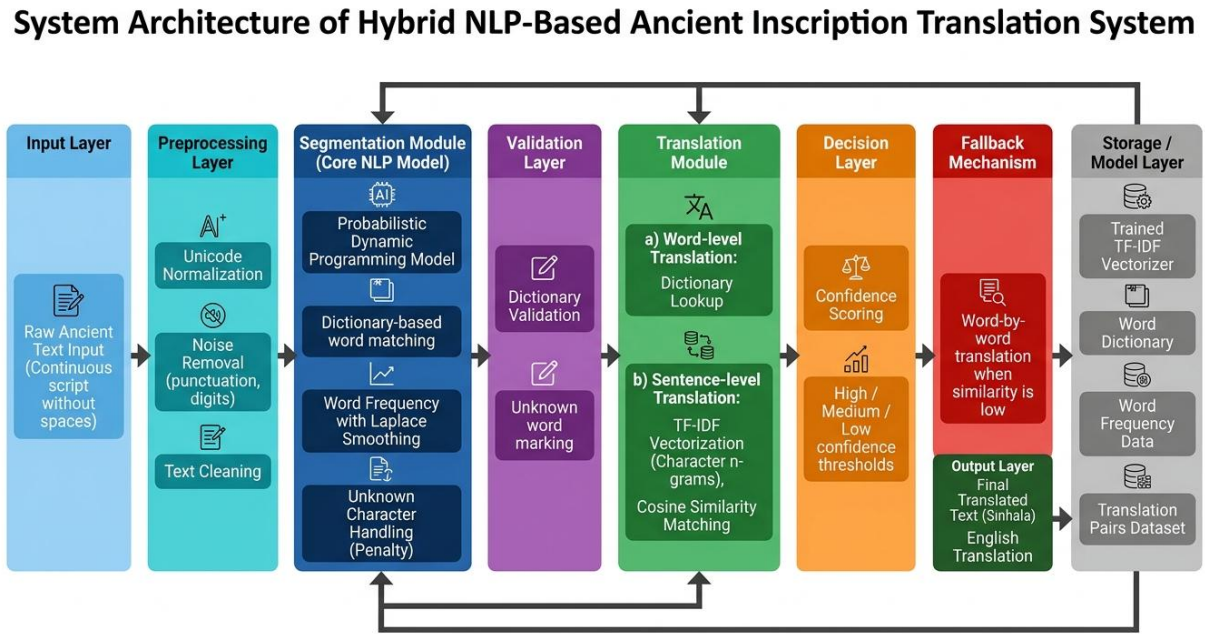

Figure 1: System Architecture Diagram

2.1.2 Data Collection and Preprocessing

The effectiveness of the proposed system depends heavily on the quality and structure of the data used. Due to the limited availability of large annotated datasets for ancient inscription text, a curated dataset was constructed using available lexical resources, manually compiled word dictionaries, and translation pairs.

In addition to secondary sources, a significant portion of the dataset was created manually to ensure domain relevance and data quality. The primary source used for this purpose was *Inscription of Ceylon, Volume I*, authored by Senarath Paranavithana. This book contains a comprehensive collection of ancient inscription records along with their interpretations in English.

Based on this source, approximately 2400 inscription sentences were manually extracted and digitized. Since the available translations in the source material were in English, an additional step was carried out to enhance the dataset for this research. Each sentence was carefully translated into Sinhala, and corresponding Sinhala meanings were also constructed to ensure consistency with the original context. This process resulted in a parallel dataset consisting of Sinhala sentence pairs and their meanings, which were specifically designed to support the translation component of the system.

Furthermore, to support the segmentation model, a separate lexical dataset was generated by splitting the constructed sentences into individual word units. This enabled the creation of a structured word dictionary along with frequency distributions, which are essential for probabilistic modeling.

The final dataset therefore consists of:

- A word dictionary containing valid lexical units derived from historical and linguistic sources
- Word frequency data used to estimate probability distributions
- Sentence-level translation pairs used for retrieval-based translation

The preprocessing stage is responsible for preparing the input text for further processing. Since ancient inscriptions often contain noise, inconsistencies, and variations in character representation, several preprocessing techniques are applied to standardize the data.

First, Unicode normalization is performed to ensure consistent encoding of characters. This step eliminates discrepancies caused by multiple representations of the same character. Next, noise removal techniques are applied using regular expressions to eliminate unwanted symbols, punctuation, and non-textual elements. This ensures that only relevant textual information is retained.

In addition, whitespace normalization is performed to handle irregular spacing, although in most cases, spaces are intentionally removed to simulate the structure of ancient inscriptions. This allows the segmentation model to operate on realistic input conditions.

Overall, preprocessing plays a crucial role in improving the reliability of subsequent stages by ensuring that the input text is clean, consistent, and suitable for probabilistic analysis.

2.1.3 Functional Requirements

Functional requirements describe the core operations and capabilities that the system must perform in order to achieve its intended objectives.

- FR1: Text Input Processing
  The system shall accept continuous text input representing recognized ancient inscription characters without predefined word boundaries.
- FR2: Text Preprocessing
  The system shall perform preprocessing operations including Unicode normalization, noise removal, and text standardization to prepare input data for further processing.

- FR3: Word Segmentation
  The system shall segment continuous input text into meaningful word units using a probabilistic dynamic programming approach based on word frequency distributions.
- FR4: Lexical Validation
  The system shall validate segmented words against a predefined dictionary and identify unknown or invalid tokens.
- FR5: Sentence Reconstruction
  The system shall reconstruct segmented words into a structured sentence format suitable for translation.
- FR6: Retrieval-Based Translation
  The system shall translate the segmented text by comparing it against a predefined corpus using TF-IDF vectorization and cosine similarity to identify the closest matching sentence.
- FR7: Confidence Evaluation
  The system shall evaluate the reliability of the translation output based on similarity scores and categorize results into confidence levels.
- FR8: Fallback Translation Mechanism
  The system shall perform word-by-word dictionary-based translation in cases where sentence-level similarity is below the defined threshold.
- FR9: Output Generation
  The system shall generate and display the final translated output along with indicators of confidence or uncertainty.

2.1.4 Non-Functional Requirements

Non-functional requirements define the quality attributes and performance characteristics of the system.

- NFR1: Accuracy
  The system shall provide accurate segmentation and translation outputs by utilizing probabilistic and statistical models optimized for low-resource data.
- NFR2: Performance Efficiency
  The system shall process input text within an acceptable time frame, ensuring efficient execution of segmentation and translation operations.
- NFR3: Scalability
  The system shall be capable of handling increasing volumes of data and expanding datasets without significant degradation in performance.

- NFR4: Reliability
  The system shall consistently produce stable and dependable outputs, even when processing noisy or incomplete input data.
- NFR5: Usability
  The system shall provide outputs in a clear and understandable format, enabling users such as researchers and historians to easily interpret the results.
- NFR6: Maintainability
  The system shall be designed in a modular manner, allowing individual components such as segmentation and translation modules to be updated or improved independently.
- NFR7: Data Integrity
  The system shall ensure that input data and processed outputs are handled accurately without loss or corruption.

2.1.5 Word Segmentation Model

Word segmentation is the core component of the system, responsible for dividing continuous text into meaningful word units. This is achieved using a probabilistic dynamic programming approach, which models segmentation as an optimization problem.

The segmentation model is based on the assumption that a valid sequence of words is more likely to occur if the individual words have higher probabilities within the language. These probabilities are estimated using word frequency distributions derived from the dataset.

Each word is assigned a probability using the following formulation:

$$P(w) = \frac{count(w) + \lambda}{total + \lambda V}$$

where:

- $count(w)$represents the frequency of the word
- $\lambda$is a smoothing factor
- $V$is the vocabulary size

To improve computational stability, logarithmic probabilities are used, allowing multiplication of probabilities to be converted into addition:

$$Score = \sum \log\ P(w)$$

The segmentation process is performed using dynamic programming, where the input text is evaluated at each position to determine the optimal split. For every substring, the algorithm considers all possible word candidates from the dictionary and computes the corresponding probability score. The segmentation with the highest cumulative score is selected as the optimal solution.

This approach is particularly effective in handling ambiguity, as it evaluates multiple possible segmentations and selects the most probable one based on statistical evidence.

In cases where no valid segmentation is found, the model applies a fallback mechanism by penalizing unknown tokens while still producing a partial segmentation. This ensures that the system remains robust even when encountering unseen or rare patterns.

From an NLP perspective, this module incorporates key concepts such as probabilistic language modeling, sequence prediction, and token boundary detection.

2.1.6 Validation Mechanism

Following segmentation, the output is validated using a lexical verification process. This step ensures that the identified words exist within the predefined dictionary and conform to expected linguistic patterns.

The validation module checks each segmented token against the vocabulary. Words that are not found in the dictionary are flagged as unknown, allowing the system to handle uncertainty more effectively. This also helps in identifying potential segmentation errors and improving the reliability of the translation stage.

This stage represents a lexicon-based NLP approach, where linguistic knowledge is used to refine and validate computational outputs.

2.1.7 Translation Model

The translation component of the system is implemented using a retrieval-based approach, which is particularly suitable for low-resource environments.

Instead of generating translations using data-intensive neural models, the system retrieves the most relevant translation from a predefined corpus based on similarity measures. This approach is both efficient and interpretable, making it ideal for ancient language processing.

The model utilizes TF-IDF (Term Frequency–Inverse Document Frequency) vectorization to represent text as numerical vectors. Unlike traditional word-based approaches, character-level n-grams are used to capture subword patterns and handle variations in spelling and structure.

The vectorization process transforms each sentence in the dataset into a weighted vector representation. When a new input is provided, it is also converted into a vector using the same transformation. Similarity between the input and stored sentences is then computed using cosine similarity:

$$Similarity = \frac{A \cdot B}{|| A ||\cdot|| B ||}$$

The sentence with the highest similarity score is selected, and its corresponding translation is returned as the output.

This approach effectively captures semantic similarity even when exact word matches are not present, making it suitable for handling noisy and inconsistent text.

2.1.8 Confidence-Based Decision System

To ensure the reliability of translation outputs, a confidence-based decision mechanism is implemented.

The system evaluates the similarity score obtained during the retrieval process and categorizes the output into three levels:

- High confidence: The similarity score exceeds a predefined threshold, and the translation is accepted directly
- Medium confidence: The translation is accepted but marked as uncertain
- Low confidence: The system applies a fallback mechanism

In low-confidence scenarios, a word-by-word translation approach is used, where each segmented word is translated individually using a dictionary. This hybrid strategy ensures that the system always produces an output, even when full sentence matching is not possible**.**

2.1.9 Tools and Technologies Used

The system is implemented using a combination of programming languages, libraries, and frameworks that support efficient NLP processing.

- Programming Language: Python
- Data Processing: Pandas, Collections
- NLP & Machine Learning: Scikit-learn (TF-IDF, cosine similarity)
- Mathematical Operations: NumPy

- Text Processing: Regular Expressions, Unicode normalization
- Model Storage: JSON, Joblib

These tools enable efficient data handling, model training, and real-time inference while maintaining scalability and flexibility.

2.1.10 Model Training and Implementation

The system follows a structured training pipeline to build segmentation and translation models.

Initially, the dataset is loaded and processed to extract word frequencies and translation pairs. The segmentation model is then constructed by computing probability distributions for each word. Simultaneously, the translation model is trained by fitting the TF-IDF vectorizer on the corpus and generating a vector space representation of all sentences.

Once trained, the models are stored for later use during inference. When a new input is provided, the system processes it through the pipeline, performing segmentation, validation, and translation in sequence.

This modular design ensures that each component can be independently updated or improved without affecting the overall system.

2.1.11 Natural Language Processing Techniques Applied

The proposed system is fundamentally built upon a combination of classical and statistical Natural Language Processing (NLP) techniques. Unlike modern deep learning approaches, which rely heavily on large-scale annotated datasets, this system adopts a hybrid methodology that is better suited for low-resource and unstructured textual data such as ancient inscriptions.

One of the primary NLP techniques utilized in this system is text normalization, which ensures that variations in character representation are standardized before processing. This is particularly important in ancient scripts where inconsistencies in encoding and transcription are common.

Another key technique is implicit tokenization, achieved through probabilistic segmentation. Since the input text does not contain explicit word boundaries, the system performs tokenization by identifying the most probable word sequences using statistical language modeling. This differs from conventional tokenization methods used in modern languages, where delimiters such as spaces are available.

The system also incorporates probabilistic language modeling, where the likelihood of a word sequence is estimated based on frequency distributions. This enables the system to make informed decisions about segmentation by selecting word combinations that are statistically more probable.

In the translation phase, the system applies vector space modeling, where text is transformed into numerical vectors using TF-IDF representation. This allows for the comparison of textual similarity using mathematical operations rather than relying on exact matches.

Additionally, information retrieval techniques are employed to identify the most relevant translation from a predefined corpus. By using cosine similarity, the system measures the closeness between the input and stored sentences, enabling accurate retrieval even in the presence of variations.

Finally, the system integrates rule-based and dictionary-based NLP methods as fallback mechanisms. These ensure that the system remains functional even when statistical confidence is low, thereby enhancing robustness and reliability.

2.1.12 System Architecture and Data Flow

The architecture of the proposed system follows a modular pipeline design, where each component is responsible for a specific stage of processing. This modularity improves maintainability, scalability, and ease of integration with other system components.

The data flow begins with the input of recognized character sequences, which are passed to the preprocessing module. This module cleans and normalizes the text before forwarding it to the segmentation engine.

The segmentation module processes the input using a probabilistic dynamic programming algorithm and outputs a sequence of candidate words. These words are then passed to the validation module, which verifies their correctness using a predefined dictionary.

The validated output is forwarded to the translation module, where the text is converted into a vector representation and compared against a corpus of known sentences. The most similar sentence is selected, and its translation is retrieved.

The final stage involves the confidence evaluation module, which determines the reliability of the translation and applies fallback mechanisms if necessary.

This structured flow ensures that errors are minimized at each stage and that the system produces consistent and interpretable outputs.

2.1.13 Algorithmic Design of Segmentation Process

The segmentation process is designed as an optimization problem, where the objective is to maximize the probability of the segmented word sequence.

The algorithm operates by iterating through the input text and evaluating all possible substrings at each position. For each substring, the system checks whether it exists in the dictionary and computes its probability score.

Dynamic programming is used to store intermediate results, allowing the system to avoid redundant computations and improve efficiency. At each step, the algorithm selects the segmentation that yields the highest cumulative score.

This approach ensures that the final segmentation is globally optimal rather than locally optimal, which is critical for handling ambiguous text.

2.1.14 Handling Ambiguity and Uncertainty

Ambiguity is a fundamental challenge in processing ancient text. A single sequence of characters can often be segmented in multiple valid ways, each leading to different interpretations.

To address this, the system employs a probabilistic approach that evaluates all possible segmentations and selects the one with the highest likelihood. This reduces the impact of ambiguity by prioritizing statistically probable solutions.

Uncertainty in translation is handled using a confidence-based mechanism. By analyzing similarity scores, the system determines whether the retrieved translation is reliable. In cases where confidence is low, fallback strategies are applied to ensure that meaningful output is still generated.

2.1.15 Justification of Selected Approach

The choice of a hybrid NLP approach is motivated by the limitations of existing methods in handling ancient inscription text.

Deep learning-based models, although powerful, require large annotated datasets and computational resources, which are not available in this domain. Furthermore, these models often lack interpretability, making it difficult to validate their outputs in a research context.

In contrast, the proposed approach offers several advantages:

- It operates effectively with limited data
- It provides interpretable results
- It is computationally efficient

- It can handle noisy and unstructured input

By combining probabilistic segmentation with retrieval-based translation, the system achieves a balance between accuracy and practicality.

2.1.16 Limitations of the Methodology

Despite its strengths, the proposed methodology has certain limitations.

The segmentation model relies on the completeness and accuracy of the dictionary. If a word is not present in the vocabulary, the system may produce incorrect segmentation.

Similarly, the translation model is limited by the size and diversity of the corpus. Since it retrieves translations based on similarity, it may not perform well for inputs that are significantly different from the training data.

Additionally, the system does not capture deep semantic relationships as effectively as neural models. However, this limitation is mitigated by the use of fallback mechanisms and confidence evaluation.

2.1.17 Summary of Methodology

The methodology presented in this research provides a comprehensive framework for processing ancient inscription text using a combination of probabilistic and statistical NLP techniques.

By integrating preprocessing, segmentation, validation, translation, and confidence evaluation into a unified pipeline, the system addresses key challenges associated with unstructured and low-resource text data.

The approach demonstrates that effective solutions can be developed without relying on large datasets or complex neural architectures, making it highly suitable for historical and linguistic applications.

2.2 Commercialization Aspects of the Product

The proposed intelligent inscription analysis system demonstrates strong commercial potential, particularly in domains focused on digital heritage preservation, linguistic research, and cultural tourism. As global interest in digitizing historical content continues to grow, there is an increasing demand for automated tools that can efficiently interpret and translate ancient inscriptions. The Word Segmentation and Translation component plays a central role in this value proposition by enabling meaningful interpretation of otherwise unreadable inscription text.

Initially, the system can be introduced through a controlled pilot phase in collaboration with academic institutions, archaeological departments, and research organizations. This phase will focus on validating the effectiveness of the segmentation and translation models in real-world scenarios, particularly when processing inscription data collected from historical sites. It will also provide an opportunity to refine the accuracy of probabilistic segmentation, evaluate the reliability of retrieval-based translation, and gather feedback from domain experts such as historians and epigraphists. These early deployments will help assess how effectively the system handles ambiguous and degraded text, as well as how well the translated outputs align with expert interpretations.

Following successful validation, the system can be scaled for broader deployment across multiple sectors. It can be offered as a software-based solution that integrates with existing research tools, digital archives, and museum information systems. Universities and research institutions can utilize the system to accelerate linguistic studies, while government bodies such as archaeological departments can adopt it to digitize and manage inscription records more efficiently. Additionally, the system can be extended to support mobile or web-based applications, enabling tourists and the general public to interactively explore historical inscriptions by obtaining instant translations.

From a commercialization perspective, the system supports multiple deployment models. It can be delivered as a cloud-based Software-as-a-Service (SaaS) platform, allowing users to upload inscription text and receive processed outputs in real time. Alternatively, it can be integrated into third-party systems through API-based services, enabling seamless incorporation into larger digital ecosystems such as heritage management platforms. The modular architecture of the system, particularly the separation of segmentation and translation components, allows for easy customization and scalability across different use cases.

The use of a hybrid NLP approach combining probabilistic segmentation and TF-IDF-based retrieval ensures that the system operates efficiently even with limited data. This makes it highly suitable for low-resource language environments, which are often overlooked by conventional AI solutions. The reliance on interpretable models also enhances trust among

domain experts, as the outputs can be validated and understood without the complexity associated with deep learning systems.

In terms of revenue generation, several monetization strategies can be adopted. The system can be licensed to institutions such as universities, museums, and archaeological departments on an annual subscription basis. API-based access can be provided to developers and research platforms for a usage-based fee. Additionally, premium features such as advanced translation capabilities, extended datasets, and analytical insights can be offered under tiered subscription plans. For large-scale users such as government organizations, customized enterprise solutions can be developed to meet specific requirements.

Furthermore, the system has the potential to expand into related domains. With additional data and model enhancements, it can be extended to support multiple ancient languages, enabling cross-cultural inscription analysis. Integration with image recognition modules can also enable a complete end-to-end solution, where inscriptions are detected, recognized, segmented, and translated within a single platform.

Overall, the proposed system presents a scalable and impactful solution that bridges the gap between historical data and modern technology. By automating the segmentation and translation of ancient inscription text, it not only enhances research efficiency but also contributes to the preservation and accessibility of cultural heritage, making it a valuable product in both academic and commercial contexts.

2.3 Testing and Implementation

2.3.1 Implementation

2.3.1.1 Word Segmentation Module

At the core of the proposed system is the word segmentation module, which is responsible for identifying meaningful word boundaries within continuous inscription text. Since ancient inscriptions do not contain explicit delimiters such as spaces, this module plays a critical role in transforming raw character sequences into interpretable linguistic units.

The segmentation process is implemented using a probabilistic dynamic programming approach. Each candidate word is assigned a probability based on its frequency within the constructed dataset. The algorithm evaluates multiple possible segmentations of the input text and selects the one with the highest cumulative probability score. This ensures that the segmentation is not only locally optimal but also globally consistent.

The dataset used to support this module was manually constructed using inscription data sourced from *Inscription of Ceylon, Volume I*. Based on this source, approximately 2400 sentences were digitized and processed. From these sentences, a word-level dictionary and

corresponding frequency distributions were generated. These frequencies were used to compute word probabilities, enabling the segmentation model to make statistically informed decisions.

To improve computational efficiency, logarithmic probabilities are used during scoring, allowing the system to handle long sequences without numerical instability. Additionally, fallback mechanisms are implemented to handle unknown words by penalizing them rather than completely discarding the segmentation path.

This module is capable of handling ambiguous text and producing reliable segmentation outputs even when the input contains noise or incomplete patterns.

2.3.1.2 Translation Module - Retrieval-Based Approach

The translation module is designed using a retrieval-based Natural Language Processing approach, which is particularly suitable for low-resource environments. Instead of generating translations using deep learning models, the system retrieves the most relevant translation from a predefined dataset based on similarity measures.

The implementation utilizes TF-IDF vectorization to convert text into numerical representations. Character-level n-grams are used instead of word-level features to capture variations in spelling and structure commonly found in ancient inscriptions. Each sentence in the dataset is transformed into a vector, creating a vector space model of the corpus.

When a segmented sentence is provided as input, it is converted into a TF-IDF vector using the same transformation. Cosine similarity is then calculated between the input vector and all vectors in the dataset. The sentence with the highest similarity score is selected, and its corresponding translation is returned as the output.

This method allows the system to identify semantically similar sentences even when exact word matches are not present. It also ensures that the translation process remains interpretable and efficient without requiring large-scale training data.

2.3.1.3 Confidence Evaluation and Fallback Mechanism

To improve the reliability of translation outputs, a confidence-based evaluation mechanism is implemented. The similarity score obtained from the retrieval process is used as an indicator of confidence.

If the similarity score exceeds a predefined threshold, the translation is accepted as a high-confidence output. If the score falls within a moderate range, the translation is provided with an indication of uncertainty. In cases where the similarity score is low, the system activates a fallback mechanism.

The fallback mechanism performs word-by-word translation using the dictionary constructed during dataset preparation. This ensures that the system is always capable of producing an output, even when sentence-level similarity is insufficient.

This hybrid approach enhances system robustness by combining statistical retrieval with rule-based translation.

2.3.1.4 Backend Processing and Model Integration

The system is implemented using Python, with modular components designed for preprocessing, segmentation, validation, and translation. Each module operates independently but is integrated into a unified pipeline.

The backend handles the flow of data between modules, ensuring that input text is processed sequentially through preprocessing, segmentation, validation, and translation stages. Intermediate outputs are stored and passed between modules, allowing for debugging and performance evaluation.

The segmentation model and translation model are trained separately and then integrated into the pipeline. The segmentation model relies on frequency-based probability distributions, while the translation model uses a trained TF-IDF vectorizer and precomputed sentence vectors.

Model persistence is achieved using JSON and serialized objects, enabling the system to load trained models efficiently during runtime.

2.3.1.5 Frontend and Output Presentation

Although the primary focus of this component is backend processing, the system provides a simple interface for displaying outputs. The segmented words and translated sentences are presented in a structured format, allowing users to easily interpret the results.

The output includes:

- Segmented word sequence
- Final translated sentence
- Confidence level of translation

This presentation ensures that users can understand both the segmentation process and the final interpretation.

2.3.2 Testing

2.3.2.1 Functional Testing

Functional testing was conducted to verify that each module of the system performs its intended operation correctly. The segmentation module was tested using manually constructed input sequences derived from the dataset. The outputs were compared against expected segmentations to ensure correctness.

The translation module was tested by providing segmented sentences and verifying whether the system retrieved the appropriate translations from the dataset. Both high-confidence and low-confidence scenarios were evaluated to ensure that the fallback mechanism was functioning as expected.

Each module was tested independently before being integrated into the full pipeline. This ensured that errors could be identified and resolved at an early stage.

2.3.2.2 Performance Testing

Performance testing was carried out to evaluate the efficiency of the system when processing input text. The segmentation algorithm was tested on varying lengths of input sequences to measure processing time and scalability.

The dynamic programming approach demonstrated efficient performance, with segmentation results generated within acceptable time limits. Similarly, the translation module was able to compute similarity scores and retrieve outputs quickly due to the use of optimized vector operations.

Overall, the system maintained consistent performance across different input sizes, demonstrating its suitability for real-time or near real-time applications.

2.3.2.3 Integration Testing

Integration testing was conducted to ensure that all modules function correctly as a complete pipeline. The system was tested using continuous input text, which was processed through preprocessing, segmentation, validation, and translation stages.

The output of each stage was verified to ensure correct data flow and consistency. The final output was evaluated to confirm that the system produces accurate and meaningful translations.

The integration tests demonstrated that the system operates cohesively, with each module contributing effectively to the overall functional.

3. RESULTS & DISCUSSION

3.1 Results

3.1.2. Word Segmentation Module

The word segmentation module forms a critical component of the proposed system, enabling the transformation of continuous inscription text into meaningful lexical units. Given that ancient inscriptions do not contain explicit word boundaries, the effectiveness of this module directly impacts the accuracy of downstream translation.

The segmentation model was evaluated using the manually constructed dataset derived from *Inscription of Ceylon, Volume I*. The dataset consisted of approximately 2400 sentences, which were used to generate both the word dictionary and frequency distributions required for probabilistic modeling. The system was tested on unseen continuous text inputs, where spaces were intentionally removed to simulate real inscription conditions.

The probabilistic dynamic programming approach demonstrated strong performance in identifying correct word boundaries across a wide range of inputs. The model was able to accurately segment frequently occurring word patterns and showed consistent behavior when handling longer character sequences. In most cases, the segmentation output closely matched manually verified ground truth sequences.

The use of frequency-based probability distributions enabled the system to prioritize more likely word combinations, reducing ambiguity in segmentation. Additionally, the use of logarithmic scoring ensured computational stability when processing longer sequences.

However, certain limitations were observed in cases involving rare or unseen words. When the input contained words not present in the dictionary, the system occasionally produced partial or suboptimal segmentation. Despite this, the fallback mechanism allowed the system to maintain continuity by generating the best possible segmentation under the given constraints.

Overall, the segmentation module demonstrated reliable performance in converting unstructured text into meaningful word units, providing a strong foundation for the translation process.

3.1.2. Translation Module (TF-IDF + Cosine Similarity)

The translation module was evaluated based on its ability to generate meaningful interpretations of segmented inscription text. The system utilizes a retrieval-based approach, where input sentences are compared against a predefined dataset using TF-IDF vectorization and cosine similarity.

The constructed dataset, which included Sinhala sentence pairs and their meanings, was used as the reference corpus for translation. Each input sentence was converted into a vector representation and compared against all stored vectors to identify the closest match.

The results showed that the system was highly effective when the input text closely matched patterns present in the dataset. In such cases, the cosine similarity scores were high, and the retrieved translations were accurate and contextually meaningful. The use of character-level n-grams allowed the system to handle minor variations in spelling and structure, improving its ability to identify semantically similar sentences.

In moderate similarity scenarios, the system was still able to produce acceptable translations, although slight variations in wording were observed. These outputs were marked with reduced confidence, allowing users to interpret them with caution.

In low similarity cases, where no close match was found, the fallback mechanism was activated. This involved translating each word individually using the dictionary. While this approach did not always produce fully fluent sentences, it ensured that the system consistently provided a meaningful interpretation rather than failing completely.

The translation module demonstrated strong adaptability to low-resource conditions, providing interpretable outputs without requiring large-scale training data.

3.1.3. Confidence Evaluation and Fallback Mechanism

The confidence evaluation mechanism played a crucial role in improving the reliability of the system outputs. By analyzing cosine similarity scores, the system was able to categorize translations into high, medium, and low confidence levels.

High-confidence outputs were observed when the input closely matched sentences in the dataset, resulting in accurate and reliable translations. Medium-confidence outputs were generated when partial similarity existed, allowing the system to provide approximate interpretations.

In low-confidence scenarios, the fallback mechanism ensured that the system remained functional. Word-by-word translation provided a basic understanding of the text, which, although less fluent, was still useful for interpretation.

This layered approach significantly enhanced system robustness and ensured consistent output generation across varying input conditions.

3.2 Research Findings

3.2.1. Effectiveness of Probabilistic Segmentation in Low-Resource Environments

The results demonstrate that probabilistic segmentation using dynamic programming is highly effective for processing ancient inscription text. Unlike rule-based approaches, the probabilistic model was able to handle ambiguity and select optimal word boundaries based on statistical evidence.

This approach proved particularly useful in low-resource environments, where large annotated datasets are not available. The ability to generate accurate segmentation using limited data highlights the suitability of this method for ancient language processing.

3.2.2. Suitability of Retrieval-Based Translation for Ancient Text

The retrieval-based translation approach showed strong performance in generating meaningful outputs without relying on complex neural models. By leveraging TF-IDF vectorization and cosine similarity, the system was able to identify semantically similar sentences even in the presence of variations.

This confirms that retrieval-based methods are a practical and efficient alternative for translation in low-resource scenarios, where traditional machine learning approaches are not feasible.

3.2.3. Importance of Dataset Construction

The manually constructed dataset played a significant role in the overall performance of the system. The inclusion of 2400 sentences and their Sinhala translations provided a strong foundation for both segmentation and translation modules.

This highlights the importance of domain-specific dataset creation, particularly in research areas where publicly available data is limited.

3.3 Discussion

3.3.1. Comparison with Existing Approaches

Compared to traditional OCR-based systems, which focus primarily on character recognition, the proposed system extends functionality by incorporating both segmentation and translation. This enables a deeper level of text interpretation, making the system more useful for real-world applications.

Unlike deep learning-based NLP models, which require large datasets and high computational resources, the proposed approach provides an efficient alternative that is both interpretable and scalable. While neural models may achieve higher accuracy in well-resourced environments, their applicability to ancient scripts remains limited.

The combination of probabilistic segmentation and retrieval-based translation offers a balanced solution that addresses both accuracy and practicality.

3.3.2. Significance of the Proposed System

The system contributes to bridging the gap between ancient inscription data and modern computational techniques. By automating the segmentation and translation process, it reduces reliance on manual expertise and improves accessibility for researchers and the general public.

The ability to process unstructured and low-resource text data demonstrates the potential of hybrid NLP approaches in specialized domains.

3.3.3. Limitations

Despite its effectiveness, the system has certain limitations. The segmentation model is dependent on the completeness of the dictionary, and missing words may lead to incorrect segmentation.

Similarly, the translation module is limited by the size and diversity of the dataset. Since the system relies on retrieval-based matching, it may struggle with inputs that are significantly different from the training data.

Additionally, the fallback translation mechanism, while useful, does not always produce grammatically complete sentences.

3.3.4. Future Research Directions

Future improvements can focus on expanding the dataset to improve both segmentation accuracy and translation quality. Incorporating additional linguistic resources and larger corpora would enhance system performance.

The integration of hybrid models that combine retrieval-based methods with lightweight neural approaches could further improve translation accuracy.

Additionally, extending the system to support multiple ancient languages and integrating it with image recognition modules would enable a complete end-to-end solution for inscription analysis.

4. SUMMARY OF INDIVIDUAL CONTRIBUTION – IT22069436

This research represents a focused and technically significant contribution to the development of an intelligent system for the analysis of ancient inscriptions, specifically through the design and implementation of the Word Segmentation and Translation component. The primary contribution of this work lies in addressing the complex challenge of interpreting continuous ancient inscription text, which lacks explicit word boundaries and structured linguistic patterns.

A key contribution of this research is the development of a probabilistic word segmentation model based on dynamic programming and frequency-based language modeling. This approach enabled the system to accurately identify meaningful word boundaries within continuous character sequences, transforming unstructured inscription text into interpretable lexical units. The use of probability distributions derived from a custom-built dataset allowed the model to handle ambiguity effectively and select the most likely segmentation paths under low-resource conditions.

Another major contribution is the implementation of a retrieval-based translation mechanism using TF-IDF vectorization and cosine similarity. This approach provided an efficient and interpretable alternative to data-intensive neural translation models, making it highly suitable for ancient languages where large parallel datasets are not available. The integration of character-level n-grams further enhanced the system's ability to handle variations in spelling and structure, improving translation accuracy.

A significant aspect of this research was the manual construction of a domain-specific dataset. Approximately 2400 sentences were extracted and digitized from *Inscription of Ceylon, Volume I*, and further processed to create Sinhala sentence pairs and corresponding meanings. This dataset served as the foundation for both segmentation and translation modules, enabling the development of a fully functional pipeline despite the limitations of publicly available resources.

In addition, the system incorporated a confidence-based evaluation mechanism and a fallback translation strategy, ensuring robustness and consistent output generation even in cases of low similarity or unknown inputs. This hybrid design allowed the system to maintain reliability while adapting to varying input conditions.

Overall, this work successfully integrates probabilistic segmentation, retrieval-based translation, and custom dataset construction into a cohesive framework, contributing toward the automation of ancient inscription interpretation. The proposed solution demonstrates both practical applicability and scalability, advancing the use of Natural Language Processing techniques in the domain of digital heritage preservation.

5. CONCLUSION

This research focused on addressing the challenges associated with the segmentation and translation of ancient inscription text, particularly in the context of low-resource languages such as early Sinhala and Brahmi. The primary objective was to develop an efficient and interpretable Natural Language Processing-based solution capable of transforming continuous, unstructured character sequences into meaningful and accessible textual interpretations.

The results of the study demonstrate that the proposed hybrid approach, combining probabilistic word segmentation with retrieval-based translation, is both practical and effective. The segmentation model successfully identified word boundaries within continuous text by leveraging frequency-based probability distributions and dynamic programming. This approach proved particularly valuable in handling ambiguity and selecting optimal segmentation paths in the absence of explicit delimiters.

The translation component further enhanced the system by providing meaningful interpretations of segmented text using TF-IDF vectorization and cosine similarity. This retrieval-based method allowed the system to generate accurate translations without relying on large-scale training data, making it highly suitable for low-resource environments. The inclusion of a fallback mechanism ensured that the system remained functional even when sentence-level similarity was insufficient, thereby improving robustness and reliability.

A significant contribution of this research lies in the manual construction of a domain-specific dataset, which enabled the development and evaluation of the system under realistic conditions. The dataset provided a strong foundation for both segmentation and translation processes, highlighting the importance of data preparation in achieving reliable results.

The overall system demonstrates the feasibility of applying classical and statistical NLP techniques to complex linguistic problems in historical domains. By automating key stages of inscription analysis, the proposed solution reduces the dependency on manual interpretation and improves accessibility for researchers, students, and the general public.

Despite its effectiveness, the system has certain limitations. The segmentation model is dependent on the completeness of the dictionary, and the translation model is constrained by the size and diversity of the dataset. Additionally, the retrieval-based approach may not fully capture deeper semantic relationships in highly complex text. These limitations provide clear directions for future improvements.

Future research can focus on expanding the dataset, integrating lightweight neural models to enhance translation accuracy, and extending the system to support multiple ancient languages. The incorporation of multimodal inputs, such as image-based inscription recognition, can also enable the development of a complete end-to-end solution.

In conclusion, this research demonstrates that meaningful interpretation of ancient inscription text can be achieved through a carefully designed hybrid NLP approach. By combining segmentation, translation, and robust fallback strategies, the system contributes toward the preservation and accessibility of historical knowledge, highlighting the potential of technology in bridging the gap between the past and the present.

6. REFERENCES

[1] Senarath Paranavithana, *Inscription of Ceylon, Volume I*, Department of Archaeology, Sri Lanka, 1970.

[2] Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[3] Jurafsky, D., & Martin, J. H., *Speech and Language Processing*, 3rd Edition, Pearson, 2023.

[4] Salton, G., Wong, A., & Yang, C. S., "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[5] Ramos, J., "Using TF-IDF to Determine Word Relevance in Document Queries," *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[6] Leskovec, J., Rajaraman, A., & Ullman, J. D., *Mining of Massive Datasets*, Cambridge University Press, 2020.

[7] Scikit-learn Developers, "TF-IDF Vectorizer Documentation," [Online]. Available: https://scikit-learn.org

[8] Bird, S., Klein, E., & Loper, E., *Natural Language Processing with Python*, O'Reilly Media, 2009.

7. GLOSSARY

AI (Artificial Intelligence):
A field of computer science focused on developing systems capable of performing tasks that normally require human intelligence, such as language understanding and decision-making.

NLP (Natural Language Processing):
A branch of AI that enables machines to understand, process, and analyze human language in both structured and unstructured forms.

Word Segmentation:
The process of dividing continuous text into individual words, especially important in languages or scripts where spaces are not explicitly used.

Dynamic Programming (DP):
An algorithmic technique used to solve optimization problems by breaking them into smaller subproblems and storing intermediate results.

TF-IDF (Term Frequency – Inverse Document Frequency):
A statistical method used to evaluate the importance of a word in a document relative to a collection of documents.

Cosine Similarity:
A mathematical measure used to determine the similarity between two text vectors based on their orientation in vector space.

Vector Space Model (VSM):
A method of representing text as numerical vectors, enabling similarity comparison and retrieval-based operations.

N-gram:
A sequence of ‘n’ characters or words used to capture patterns in text. In this research, character-level n-grams are used to handle spelling variations.

Corpus:
A collection of text data used for training and evaluation in NLP tasks.

Tokenization:
The process of breaking text into smaller units such as words or phrases.

Low-Resource Language:
A language with limited available digital data and annotated datasets for computational processing.

Fallback Mechanism:
A backup method used when the primary system fails to produce reliable output, such as word-by-word translation.

Unicode Normalization:
A process used to standardize character encoding, ensuring consistency across text inputs.

Dictionary-Based Translation:
A simple translation method where individual words are mapped directly to their meanings using a predefined dictionary.