*International Journal of Basic and Applied Sciences*, 14 (5) (2025) 187-200

**International Journal of Basic and Applied Sciences**

*Website: www.sciencepubco.com/index.php/IJBAS*
*https://doi.org/10.14419/rs62f777*
***Research paper***

# Evaluating Machine Learning and Deep Learning Techniques for Part-Of-Speech Tagging in Tamil

**Dr. N. Kannaiya Raja [1] *, Dr. Pawan Kumar Chaurasia [2], Prof Dr. Midhunchakkaravarthy [3]**

*[1] Post Doctoral Researcher, Lincoln University College, Malaysia*
*[2] Babasaheb Bhimrao Ambedkar Central University, Lucknow, Uttar Pradesh, India*
*[3] Prof Dr Midhunchakkaravarthy, Lincoln University College, Malaysia*
**Corresponding author E-mail: pdf.kannaiya@lincoln.edu.my*

*Received: July 25, 2025, Accepted: August 30, 2025, Published: September 6, 2025*

## Abstract

Part-of-speech (POS) tagging for Tamil is a importance task due to the language's highly inflectional and agglutinative morphology. This study systematically evaluates both machine learning and deep learning models including Conditional Random Fields (CRF), Support Vector Machine (SVM), Hidden Markov Model (HMM), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), and LSTM-RNN with CRF output for Tamil POS tagging, using a well-annotated CLE-style benchmark dataset. We employed a comprehensive, lanlanguage-independent feature set and performed 10-fold cross-validation to ensure robust results. Experimental finding that, for the CLE da-dataset, the CRF model achieves the highest average accuracy at 86.32%, outperforming SVM (81.13%), LSTM-RNN (78.64%), LSTM-RNN-CRF (78.03%), and HMM (78.03%). In contrast, on the more challenging BJ dataset, the LSTM-RNN deep learning model attains the highest accuracy of 92.70%, followed closely by CRF (91.2%), LSTM-RNN-CRF (91.02%), HMM (90.11%), and SVM (86.25%). These results highlight the importance of model selection in morphologically rich languages, while CRF is optimal for structured and moderately sized datasets, LSTM-RNN deep learning approaches excel on larger. This work establishes new empirical benchmarks for Tamil POS tagging and demonstrates that advanced neural models provide a clear advantage in handling Tamil's linguistic complexity.

***Keywords***: *Tamil; Part-of-Speech Tagging, POS; CRF; LSTM-RNN; Machine Learning; Deep Learning, NLP; Agglutinative Language; Accuracy.*

## 1. Introduction

Part of Speech (POS) tagging is a fundamental task in natural language processing (NLP) that involves assigning appropriate syntactic tags to each word in a given text. This task is typically achieved using taggers, which rely on a set of linguistic rules or algorithms designed to classify words into predefined categories such as nouns, verbs, adjectives, and others. The effectiveness of a POS tagging system [1] directly influences the accuracy of various NLP applications, including speech recognition, machine translation, information extraction, and text-to-speech systems.

The performance of POS tagging models depends significantly on two primary factors: (1) the size and quality of the training dataset, and (2) the design of the tagset used for annotation. A well-structured training dataset with balanced representation of linguistic features, coupled with an effective tagset, is essential for developing robust POS tagging systems. This task becomes even more challenging in languages with fewer linguistic resources, such as Tamil, which is morphologically rich and highly inflectional. The scarcity of annotated datasets and the complexity of the language pose significant hurdles in achieving accurate POS tagging for Tamil[2].

NLP has become a pivotal domain in the information year, where interpreting complex language patterns is a cornerstone of artificial intelligence. Machine learning (ML) and deep learning (DL) have emerged as transformative approaches in NLP, enabling automated systems to achieve state-of-the-art performance. While ML techniques like Support Vector Machines (SVM) and Hidden Markov Models (HMM) have traditionally been employed in POS tagging, the recent advancements in deep learning, particularly in Recurrent Neural Networks (RNN) [3]and their variants, have demonstrated superior performance across a range of NLP tasks. Deep learning has not only revolutionized NLP but also significant impact in fields like image processing, machine translation, and bioinformatics, showcasing its adaptability and robustness[4].

Motivated by the success of deep learning models, this study evaluates both machine learning and deep learning approaches for POS tagging for the Tamil language. The research includes the implementation of Conditional Random Fields (CRF), SVM, HMM, and two deep learning models, such as Long Short-Term Memory (LSTM)-RNN and LSTM-RNN with CRF output. CRFs are particularly suited for sequence labeling tasks as they model relationships between adjacent items in a sequence, making them a natural choice for POS tagging. By leveraging both language-dependent and language-independent features, the study aims to design a novel set of features work which is tailored for Tamil, address the unique challenges associated with Tamil POS tagging, and explore the comparative performance of machine learning and deep learning models.

The objective of this work is (a) to analyze the challenges associated with Tamil POS tagging, (b) to design an optimized feature set based on context word windows, and (c) to evaluate and compare the performance of machine learning and deep learning models, thereby establishing a benchmark for future research in Tamil POS tagging. The rest of the paper is structured as follows Section 2 reviews related work on Tamil POS tagging, Section 3 discusses the linguistic challenges in Tamil, Section 4 provides an overview of the CRF, SVM, RNN, and HMM models, Section 5 details the experimental setup and evaluation, Section 6 presents a discussion and error analysis, and Section 7 concludes the study with future directions.

## 2. Related Work

The approaches adopted for Tamil Part of Speech (POS) tagging can be broadly categorized into three groups are rule-based, statistical, and machine learning-based methods. Each category has its own strengths and challenges when applied to Tamil, a morphologically rich and highly inflectional language.

### 2.1. Rule-Based Approaches

The initial efforts in Tamil POS tagging were primarily rule-based. These methods relied on manually crafted linguistic rules and grammatical constructs of Tamil to assign POS tags to words. One such early attempt was by Baskaran et al., where the authors developed a rule-based system using handcrafted rules derived from Tamil syntax and morphology. However, despite achieving reasonable accuracy, rule-based approaches have significant limitations, including difficulty in scaling to large datasets, inflexibility in handling ambiguous or unseen data, and the need for extensive domain expertise in Tamil grammar and Additionally, these approaches are time-consuming and challenging to adapt to evolving linguistic patterns, making them less practical for large-scale NLP tasks[5].

### 2.2. Statistical Approaches

The next evolution in Tamil POS tagging involved statistical methods, which addressed some of the limitations of rule-based systems by leveraging probabilistic models. One of the early statistical methods for Tamil POS tagging was based on n-gram Hidden Markov Models (HMM). This approach utilized the probabilistic relationships between words and tags, achieving improved accuracy over rule-based methods. However, statistical models require a large corpus of annotated data for training, which has been challenging for Tamil due to the limited availability of high-quality annotated datasets [6].

### 2.3. Machine Learning Approaches

Supervised machine learning techniques have emerged as a popular choice for Tamil POS tagging due to their ability to learn from annotated data and generalize well to unseen text. Support Vector Machines (SVM) and Conditional Random Fields (CRF) have been extensively studied for POS tagging in Tamil. For example, Kannan et al. employed CRFs with both language-dependent and language-independent features to address the POS tagging challenge in Tamil. Their model demonstrated improved performance compared to rule-based and statistical methods, particularly in handling complex morphological [7] patterns.

### 2.4. Deep Learning Approaches

In recent years, deep learning models have shown significant promise in POS tagging tasks across multiple languages, including Tamil. Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks, are particularly effective in sequence labeling tasks like POS tagging. These models can capture long-term dependencies in text, making them well-suited for morphologically rich languages like Tamil. For instance, recent studies have explored the use of LSTM networks with CRF output layers, achieving state-of-the-art performance on Tamil POS tagging benchmarks. Deep learning models also benefit from pre-trained word embeddings, such as Fast Text and BERT, which provide rich semantic representations of words[8].

### 2.5. Challenges in Tamil POS Tagging

Despite advancements in machine and deep learning approaches, Tamil POS tagging faces unique challenges due to the language's agglutinative nature, the extensive use of compound words, and the limited availability of annotated corpora. Additionally, designing an effective tagset and feature set that can handle Tamil's rich morphology and syntactic variations remains a non-trivial task.

### 2.6. Summary of Existing Approaches

Table 1 provides a summary of the various approaches adopted for Tamil POS tagging, highlighting the methods, datasets, and reported accuracies. These studies collectively underscore the need for robust models that can effectively address Tamil's linguistic complexities while leveraging advancements in machine learning and deep learning.

In this study, we build upon these prior works by implementing and comparing machine learning (CRF, SVM, HMM) and deep learning (LSTM-RNN, LSTM-RNN with CRF) models for Tamil POS tagging. Our objective is to evaluate the effectiveness of these approaches in addressing the unique challenges of Tamil POS tagging and establish a benchmark for future research. Recent advancements in machine learning and deep learning have significantly influenced Part-of-Speech (POS) tagging for the Tamil language, a morphologically rich and agglutinative language. This section reviews notable studies from 2024 that have contributed to this field.

### 2.7. Deep Learning Models for Tamil POS Tagging

Visuwalingam et al. proposed a deep learning-based POS tagger utilizing Bidirectional Long Short-Term Memory (BLSTM) networks. Their model achieved an accuracy of 99.83% for known words and 95.03% for unknown words by integrating word-level, character-level, and pre-trained word embeddings[9].

While Visuwalingam et al. (2024) report exceptionally high accuracies (99.83% for known words, 95.03% for unknown words), such figures may reflect the specific dataset design, limited domain scope, or differences in experimental protocols. These results, while impressive, may not be directly comparable to our CLE and BJ experiments, which are larger and morphologically diverse. Therefore, we reference their findings for completeness but caution against direct numerical comparison without standardized evaluation conditions.

**Table 1:** Summary of Previous Tamil POS Approaches

| Approach | Methods | Datasets | Accuracy |
|---|---|---|---|
| Rule-Based | Handcrafted Rules | Small Corpus of Tamil Text | ~80% |
| Statistical | Hidden Markov Models (HMM) | Annotated Tamil Corpus | ~85% |
| Machine Learning | Support Vector Machines (SVM), Conditional Random Fields (CRF) | Tamil POS Tagged Dataset | ~90% |
| Deep Learning | Recurrent Neural Networks (RNN), LSTM | Tamil POS Benchmark Dataset | ~95% |

### 2.8. Hybrid Deep Learning Architectures

Aravinthan and Eugene explored hybrid deep learning architectures combining Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Units (Bi-GRU) for Tamil text classification. Their models outperformed traditional machine learning approaches and multilingual BERT models, highlighting the effectiveness of hybrid architectures in handling Tamil's linguistic complexities[10].

### 2.9. Neural-Based POS Taggers

ThamizhiPOSt, which was developed by Sarveswaran, is a neural-based POS tagger for Tamil built using the Stanza framework and trained on 11,000 POS-tagged sentences. It employs the Universal Dependency POS tagset and achieves an accuracy of 95.20% on the Tamil Dependency Treebank[11].

### 2.10. Challenges in Tamil POS Tagging

Despite these advancements, Tamil POS tagging remains challenging due to the language's agglutinative nature, extensive use of compound words, and limited availability of annotated corpora. Designing effective tagsets and feature sets that accommodate Tamil's rich morphology and syntactic variations continues to be a complex task.

These studies collectively underscore the progress made in applying machine learning and deep learning techniques to Tamil POS tagging. They also highlight the ongoing challenges and the need for further research to develop robust models capable of effectively addressing Tamil's linguistic complexities.

### 2.11. Transformer-Based Approaches

In addition to classical approaches such as CRF, SVM, and HMM, and neural architectures like LSTM, recent research in NLP has increasingly relied on transformer-based models such as BERT, mBERT, IndicBERT, and Tamil-specific pre-trained models. These architectures leverage self-attention mechanisms to capture long-range dependencies more effectively than recurrent models. In the context of Tamil POS tagging, transformer models have demonstrated significant improvements by incorporating contextual embeddings at both the word and sub-word levels. Although not evaluated in this study, the inclusion of transformers in future work is crucial, as they currently represent the state of the art for sequence labeling tasks. Their integration would provide deeper insights into addressing Tamil’s agglutinative morphology, code-mixed texts, and low-resource scenarios.

The literature on Tamil POS tagging has evolved from early rule-based and statistical systems (e.g., Brill, 1995; Kannan et al., 2006) to more advanced machine learning and neural architectures (e.g., Sarveswaran, 2022; Visuwalingam et al., 2024). This progression ensures both historical depth and contemporary relevance. However, despite these advances, the lack of standardized evaluation across datasets and the limited exploration of Tamil’s morphological complexity highlights a clear research gap.

Our study addresses this gap by systematically comparing multiple machine learning and deep learning models on benchmark datasets, thereby providing a unified perspective on the current state of Tamil POS tagging. In doing so, it not only builds upon prior research but also critically analyzes the limitations of rule-based and statistical approaches. Furthermore, while recent advancements such as pre-trained embeddings (e.g., FastText, BERT) have shown promising results, their role could be emphasized further as they represent an important direction for enhancing Tamil POS tagging.

## 3. Tamil Language

Tamil, one of the classical and Dravidian languages, is known for its rich morphology and agglutinative nature. Unlike languages scripted from right to left, Tamil is written from left to right using the Tamil script, which comprises 12 vowels, 18 consonants, and a unique set of 216 combined vowel-consonant characters. Tamil is syntactically context-sensitive, with the form and structure of words being heavily influenced by their position and context within a sentence.

A key linguistic characteristic of Tamil is its agglutinative morphology, where words are formed by attaching affixes to a root word. These affixes denote grammatical features such as tense, case, number, and gender, making Tamil highly inflectional. For instance, verbs in Tamil can have multiple suffixes indicating person, number, tense, and mood, creating a vast array of word forms. Similarly, Tamil nouns are modified through extensive case suffixes, further enriching the language's morphological complexity.

Tamil's free word order nature adds to the challenges of natural language processing (NLP). While the default sentence structure is Subject-Object-Verb (SOV), the order of words can change without altering the semantic meaning of the sentence. This flexibility, while enriching the expressiveness of Tamil, poses difficulties in tasks like POS tagging and syntactic parsing, as the syntactic structure does not always align with standard rules.

Other characteristics that make Tamil a challenging language for NLP tasks include:

1) Compound Words: Tamil frequently uses compound words that merge multiple lexical items, requiring advanced segmentation techniques.
2) Sandhi Rules: Phonetic changes occur at word boundaries due to morphophonemic alterations.

3) Lack of Capitalization: Tamil does not have a concept of capitalization, making it harder to identify proper nouns or the beginning of sentences.
4) Loan Words: Tamil incorporates vocabulary from Sanskrit, English, and other languages, which adds to the linguistic diversity and complexity.
5) Limited Annotated Resources: Despite being a widely spoken language, Tamil suffers from a lack of high-quality annotated corpora for NLP tasks like POS tagging.

The first POS tagset for Tamil was developed as part of early computational linguistic research and has since evolved to include tagsets compatible with modern NLP frameworks like Universal Dependencies (UD). These tagsets are designed to balance linguistic richness with usability for computational models. However, the absence of standardized linguistic resources and the vast morphological diversity of Tamil still pose significant challenges for developing robust NLP systems.

The development of Part-of-Speech (POS) tagsets for the Tamil language began to emerge in the early 2000s. Initially, small-scale tagsets were created based on traditional Tamil grammar rules. These early tagsets included only a limited number of categories, which led to challenges in performing Natural Language Processing (NLP) tasks effectively due to a lack of granularity and coverage.

In response, around 2004, the Anna University NLP Lab, in collaboration with the AU-KBC Research Centre, developed an extensive tagset comprising approximately 45 POS tags. This tagset provided a detailed classification of categories such as nouns, verbs, particles, adjectives, and grammatical markers. However, some categories overlapped or were redundant, which introduced complexity and ambiguity during computational processing.

Subsequently, in 2010, the Tamil Treebank Project introduced a refined POS tagset consisting of 34 tags, aimed at resolving the issues found in earlier versions. This newer tagset offered better classification of verbs and their tenses, and attempted to map the syntactic structure more precisely. While the set aligned well with Tamil grammatical rules, it did not fully address distinctions in categories such as particles and adjectives.

More recently, institutions like IIIT-Hyderabad and the EMILLE Corpus Project jointly released a Universal POS Tagset for Tamil, which featured 12 major categories and 28 subcategories. Modeled similarly to the CLE (Centre for Language Engineering) tags et in Urdu, this structure was designed with both linguistic precision and computational efficiency in mind. In creating this tagset, widely accepted standards such as the Penn Treebank and Indian language POS tagging guidelines were used as references.

Over time, POS tagsets for Tamil have undergone significant evolution. Each newer version has aimed to overcome the limitations of previous models through careful linguistic analysis and syntactic coherence. As a result, current tagsets like the Universal and CLE-inspired models are considered more suitable and practical for modern NLP tasks in Tamil, offering a balanced trade-off between linguistic richness and computational tractability. The Tamil POS tagset classifies words into main grammatical categories along with their relevant subcategories to support accurate linguistic analysis. It includes பெயர்ச்சொல் (Nouns) like proper and common nouns, சுட்டுப்பெயர்கள் (Pronouns) such as personal, relative, and adverbial forms, and தெரிவுச்சொற்கள் (Demonstratives) categorized as adjectival, relative, or personal. உதவிசொற்கள் (Auxiliaries) are divided into aspectual and tense types, while எண்கள் (Numbers) include cardinal, ordinal, fractional, and multiplicative forms. இணைப்புச்சொற்கள் (Conjunctions) are tagged as coordinating or subordinating, and தலைப்புகள் (Titles) are split into pre- and post-titles. Several other grammatical markers like வினையச்சொல் (Verb), வினையெச்சம் (Adverb), Reflexives, Phrase Markers, Quantifiers, and Date/Expression words are also included, though without subcategories. This tagset offers a robust structure for computational processing and syntactic analysis of Tamil text[11].

The Word file titled "Tamil_CLE_Style_POS_Tagset.docx" presents a structured table that outlines a Tamil adaptation of the CLE-style Part-of-Speech (POS) tagset, inspired by the Centre for Language Engineering (CLE) framework developed for Urdu and other languages. Designed for linguistic analysis and NLP applications, the tagset includes 12 main POS categories in Tamil—such as பெயர்ச்சொல் (Noun), சுட்டுப்பெயர் (Pronoun), வினைச்சொல் (Verb), உதவிசொல் (Auxiliary), வினையெச்சம் (Adverb), மீதமுள்ளவை (Residual), உரையினைச்சொல் (Interjection), இடப்பெயர் (Adposition), சின்னங்கள் (Symbols), துணைச்சொல் (Particle), இணைப்புச் சொல் (Conjunction), and பெயர்ச் சேர்மொழி (Nominal Modifiers)—and 35 subcategories, which include specific linguistic forms like சிறப்புப்பெயர் (Proper Noun), பொதுப்பெயர் (Common Noun), தனிப்பட்ட சுட்டுப்பெயர் (Personal Pronoun), காட்டு சுட்டுப்பெயர் (Demonstrative), உரிமை சுட்டுப்பெயர் (Possessive), முதன்மை வினை (Main Verb Infinitive), முடிவடைந்த வினை (Finite Verb), and various auxiliaries including aspectual, progressive, tense, and modal types. Additional distinctions are made for adverbial types such as எதிர்மறை (Negation), structural particles like வாலா (VALA), and modifiers such as வரிசை (Ordinal), பின்னிப்பு (Fractional), and அளவீடு (Quantifier). Developed with reference to resources like the Penn Treebank and Indian language POS tagsets, this Tamil CLE-style framework supports syntactic annotation, POS tagging, parsing, and machine translation, while being carefully aligned with Tamil's morpho-syntactic characteristics for effective use in academic research and computational linguistics.

## 4. Tamil Part-of-Speech Tagging Challenges

Early Tamil grammatical frameworks, rooted in classical grammar like Tolkappiyam, broadly classified words into major categories such as Peyarchol (Noun), Vinaichol (Verb), and Idaichol (Particles/Connectives). However, modern computational linguistics for Tamil recognizes the need for finer-grained distinctions, often encompassing 12 or more morpho-syntactic categories as seen in recent tagsets like those inspired by CLE and Indian language Treebanks. Tamil, being a morphologically rich and agglutinative language, presents unique challenges for POS tagging, similar to Urdu and other Indian languages. One of the fundamental issues is that a single root word in Tamil can yield numerous inflected and derived forms, often depending on case, tense, number, gender, and mood. For instance, Tamil verbs conjugate for tense, aspect, voice, polarity, person, number, and gender, resulting in potentially hundreds of forms for a single root. Similarly, Tamil nouns inflect for case markers (எடைச் சுட்டிகள்) and postpositions, complicating their classification in syntactic structures[12].

**Table 1:** POS Tagset for Tamil Language.

| Main Tag (Tamil) | Sub Categories (Tamil) |
|---|---|
| பெயர்ச்சொல் (Noun) | சிறப்புப்பெயர் (Proper Noun), பெயர்ச்சொல் (Noun) |
| சுட்டுப்பெயர் (Pronoun) | தனிப்பட்ட சுட்டுப்பெயர் (Personal pronoun), சுட்டுப்பெயர் (Relative), உரிச்சொல் (Adverbial), KAF, உரிச்சொல் KAF (AKP) |
| தெரிவுச்சொல் (Demonstrative) | விளக்கச்சொல் (Adjectival), சுட்டுப்பெயர் (Relative), தனிப்பட்ட (Personal) |
| உதவிசொற்கள் (Auxiliaries) | நிலைவியல் உதவிசொல் (Aspectual Auxiliaries), கால உதவிசொல் (Tense Auxiliaries) |
| எண்கள் (Number) | எண்கள் (Cardinal), வரிசை (Ordinal), பின்னிப்பு (Fractional), பலமடங்கு (Multiplicative) |
| இணைப்புச்சொல் (Conjunction) | இணைப்பு (Coordinative), சார்பியல் (Subordinating) |
| தலைப்பு (Title) | முன்தலைப்பு (Pre-title), பிந்தலைப்பு (Post-title) |
| வினையச்சொல் (Verb) | --- |
| SE/SE | --- |
| வினையெச்சம் (Adverb) | --- |
| Genitive | --- |
| Genitive (V) | --- |
| Reflexive | --- |
| சொற்றொடர் குறி (Phrase Marker) | --- |
| Adjectival Practical(A) | --- |
| தேதி (Date) | --- |
| வெளிப்பாடு (Expression) | --- |
| Quantifier | --- |
| VALA (வாலா) | --- |
| அளவீட்டு அலகு (Measuring Unit) | --- |
| Question Word | --- |
| Negative (மறு) | --- |
| Sentence Marker | --- |
| உரையினைச்சொல் (Interjection) | --- |
| தீவிரம் (Intensifier) | --- |
| KER | --- |

Tamil also lacks capitalization, which hinders the identification of proper nouns, and like other Dravidian languages, exhibits free word order, making dependency parsing and contextual disambiguation difficult. Furthermore, pronouns and demonstratives often function as adjectives, depending on their placement in the sentence, leading to tagging ambiguities. For example, the word "எந்த" (which) can be a determiner in one context and an interrogative pronoun in another. Similarly, nouns often act as adjectives, especially when the head noun is dropped, as in "பள்ளி மாணவர்கள்" (school students) vs "பள்ளி வந்தது" (the school came), where “பள்ளி” shifts its function.

Another common challenge is adjectives used as adverbs, like "அருகில் வந்தான்" (came near), where "அருகில்" (near) may be tagged as an adverb though it is noun-derived. Similarly, adjectives acting as nouns, such as "பழையவர்" (elder/old one) derived from "பழைய" (old), blur categorical boundaries. Loanwords and code-mixed Tamil-English phrases common in modern usage and social media further complicate tagging due to a lack of standard dictionary entries. Moreover, Tamil lacks large annotated corpora, domain-specific lexicons, and high-quality morphological analyzers, which are essential for training accurate POS taggers. Most errors in automatic Tamil POS tagging occur while distinguishing proper nouns from common nouns, adjectives from participles, and case markers from postpositions. Without sufficient syntactic cues or capitalization, relying on local and global context becomes critical, demanding more advanced models such as context-aware deep learning or transformer-based taggers[13].

While both CLE and BJ corpora are widely used benchmarks for Tamil POS tagging, they differ substantially in size, complexity, and domain. The CLE dataset contains ~145K words with 4,820 sentences across 88 documents, focusing on more structured and balanced text. In contrast, the BJ dataset is larger and more diverse, comprising ~200K+ tokens with a wider distribution of syntactic categories, including higher frequencies of verbs and postpositions. The BJ dataset contains a higher degree of morphological variation, for example, case markers, verb inflections, compound words, along with informal constructions that do not strictly follow standard grammar. Such linguistic richness increases the complexity of identifying correct POS tags. Deep learning models like LSTM-RNN are well-suited for this scenario because they can capture long-range dependencies and learn contextual patterns beyond local features. In contrast, the CLE dataset is more structured and moderately scaled, making it easier for feature-based models like CRF to exploit its regularity. Thus, the contrast in dataset characteristics directly explains why LSTM-RNN dominates on BJ, while CRF remains effective on CLE.

**Table 2:** The Description of POS Tagset for Tamil Language

| Main Tags (Tamil) | Sub Categories (Tamil) |
|---|---|
| பெயர்ச்சொல் (Noun) | சிறப்புப்பெயர் (NNP), பொதுப்பெயர் (NN) |
| சுட்டுப்பெயர் (Pronoun) | தனிப்பட்ட சுட்டுப்பெயர் (PRP), காட்டு சுட்டுப்பெயர் (PDM), உரிமை சுட்டுப்பெயர் (PRS), எதிரொலிப் பெயர் (PRF), சொந்தக் காட்டு (APNA), சார்ந்த சுட்டுப்பெயர் (PRR), சார்ந்த காட்டு (PRD) |
| வினைச்சொல் (Verb) | முதன்மை வினை (VBI), முடிவடைந்த வினை (VBF) |
| உதவிசொல் (Auxiliary) | நிலைவியல் உதவிசொல் (AUXA), முன்னேற்ற உதவிசொல் (AUXP), கால உதவிசொல் (AUXT), விதி உதவிசொல் (AUXM) |
| மீதமுள்ளவை (Residual) | வெளிநாட்டு துணைச்சொல் (FF) |
| உரையினைச்சொல் (Interjection) | உரையினைச்சொல் (INJ) |
| இடப்பெயர் (Adposition) | முன்னணிப் பெயர்ச்சொல் (PRE), பின்வைக்கை (PSP) |

| | |
|---|---|
| சின்னங்கள் (Symbols) | பொது குறியீடு (SYM), முடிவுக்குறி (PU) |
| வினையெச்சம் (Adverb) | பொதுவினையெச்சம் (RB), எதிர்மறை (NEG) |
| துணைச்சொல் (Particle) | பொது துணைச்சொல் (PRT), வாலா (VALA) |
| இணைப்புச் சொல் (Conjunction) | இணைப்பு (CC), சார்பு இணைப்பு (SC), எஸ்.சி கார (SCK), முன்னிலை (SCP) |
| பெயர்ச் சேர்மொழி (Nominal Modifiers) | வரிசை (OD), பின்னிப்பு (FR), பலமடங்கு (QM), சிறப்பு பெயரடை (JJ), அளவீடு (Q), அடிப்படை எண் (CD) |

## 5. Methodology

This section discusses the various models tested in the context of Part-of-Speech (POS) tagging for the Tamil language. The models evaluated in this study include Conditional Random Fields (CRF), Support Vector Machines (SVM), Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN), BiLSTM-CRF hybrid architectures, and a traditional N-gram Language Model. Given the morphologically rich and agglutinative nature of Tamil, each of these models presents unique advantages and challenges when applied to sequential tagging tasks[14].

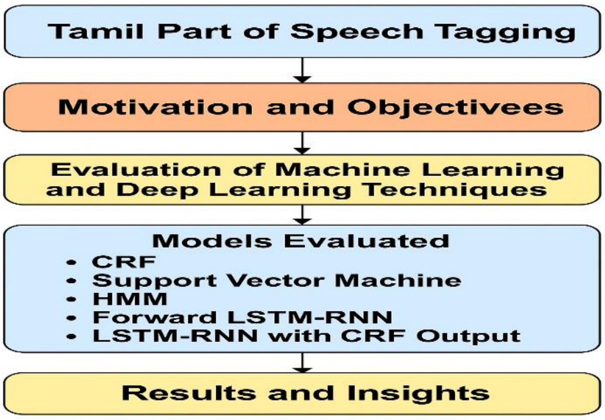
Tamil Part of Speech Tagging
Motivation and Objectivees
Evaluation of Machine Learning and Deep Learning Techniques
Models Evaluated
• CRF
• Support Vector Machine
• HMM
• Forward LSTM-RNN
• LSTM-RNN with CRF Output
Results and Insights

**Fig. 1:** Research Flow Diagram.

A. Conditional Random Fields

Conditional Random Fields (CRFs) are widely adopted probabilistic models used for sequence labeling and segmentation tasks in Natural Language Processing (NLP), including POS tagging. CRFs treat the POS tagging problem as a structured prediction task, where the goal is to find the most likely sequence of tags Y = ($y_1$, $y_2$, ..., $y_n$) given an observation sequence X = ($x_1$, $x_2$, ..., $x_n$), where each $x_i$ represents a token in the input Tamil sentence[15].

CRFs define a conditional probability distribution over possible label sequences given an input sequence, represented mathematically as:

$$P(Y \mid X) = \frac{1}{Z(X)} \exp(i\sum j\sum \lambda jtj(yi - 1, yi, X, i) + k\sum \mu ksk(y, X, i)) \quad (1)$$

Where:

- $t_j$ represents transition feature functions (i.e., dependencies between tags).
- $s_k$ denotes state feature functions (i.e., features of the observation at position i).
- $\lambda_j$ and $\mu_k$ are learned weights for features during training.
- Z(X) is a normalization term ensuring that the sum of probabilities over all tag sequences equals 1.

CRFs are particularly well-suited for Tamil POS tagging because they capture contextual dependencies and learn sequential patterns while allowing for rich, handcrafted features such as suffixes, prefixes, part-of-speech context windows, and character n-grams, all of which are vital for morphologically rich languages like Tamil. Moreover, CRFs maintain label consistency by enforcing sequential constraints, which are often violated in simple classifiers[16].

B. Feature functions

Feature functions are integral components of the CRF model's training process. They are binary-valued functions that return either 1 or 0 which based on the presence or absence of specific lexical or contextual patterns. These functions guide the CRF in learning transitions between output classes (POS tags) by considering both the current word and its context (e.g., previous or next words). In this study, we use a context window size of 7 words and 35 output POS tags based on the CLE-style tagset.

Each feature function evaluates a particular condition, such as the left context word being a specific Tamil word or the current word having a particular suffix, and is activated if the condition is met. Below is an example of how these functions are structured in Tamil, analogous to Table 3.

**Table 3:** Feature Functions

| Feature Function | Condition | Example | Return |
|---|---|---|---|
| func1 | if (output class = NN and left context word = "ஒரு") | Current word: "மாணவர்", Left word: "ஒரு" | return 1 else 0 |
| func2 | if (output class = JJ and suffix of current word = "ஆன") | Current word: "பெரிய", Suffix: "ஆன" | return 1 else 0 |
| func3 | if (output class = VB and previous word = "அவன்") | Current word: "பாடுகிறான்" | return 1 else 0 |
| funcN | if (output class = PRP and current word = "நான்") | Current word: "நான்" | return 1 else 0 |

In Tamil POS tagging using CRF, feature function size estimation plays a crucial role in capturing linguistic patterns across a given context. With a context window of 7 tokens and 35 output POS classes, the number of unigram template feature functions generated for a single token is calculated as (1 × 7) × 35 = 245. For a sentence consisting of 12 tokens, the total number of binary-valued feature functions becomes (12 × 7) × 35 = 2,940. These functions are automatically generated during the training phase and evaluate specific conditions based on lexical and contextual clues in Tamil. For example, words ending with suffixes such as "து" or "அன்" may indicate verbal nouns or masculine pronouns, respectively, and can be encoded as feature conditions to improve tagging accuracy. Such extensive feature generation enables the CRF model to learn intricate morpho-syntactic dependencies inherent to the Tamil language.

### C. Support Vector Machine (SVM)

Support Vector Machines (SVMs) are supervised machine learning algorithms primarily designed for binary classification tasks. SVMs operate by transforming input data into a higher-dimensional space using nonlinear mapping, where they then construct an optimal hyperplane that maximally separates classes. In the domain of Natural Language Processing (NLP), SVMs have been effectively applied to tasks such as Part-of-Speech (POS) tagging, Named Entity Recognition (NER), segmentation, and document classification, offering strong generalization capabilities and high accuracy without overfitting[17].

In Tamil POS tagging, the use of SVMs has proven promising due to the model's capacity to handle sparse feature spaces and morphologically rich structures. Let the training dataset be defined as

$$(x_i, y_i)_i^n = 1 \tag{2}$$

where

$$x_i \in R^d \tag{3}$$

Denotes the input feature vector derived from a Tamil token and its context, and

$$y_i \in \{-1, +1\} \tag{4}$$

Represents the POS tag label associated with $x_i$. The primary objective in SVM is to identify a decision boundary using a linear discriminant function:

$$f(x) = w \cdot x + b \tag{5}$$

Here, w is the weight vector, x is the input vector, and b is the bias. This function assigns a score to x and determines its class based on the sign of the score. For a binary classification task, points satisfying

w·x+b>0 are classified into one class, while those. For POS tagging in Tamil, where the number of tags exceeds two, multi-class classification is achieved by combining multiple binary classifiers using one-vs-one or one-vs-all schemes.

In higher-dimensional mapping, the kernel trick is used to replace the dot product with a kernel function K(x ,z), which maps the input to a nonlinear feature space. The discriminant function becomes:

$$f(x) = \text{sign}(\sum_{i=1}^{\infty} w_{i\ \mathrm{k}(x,y)}\ +\ b) \tag{6}$$

$z_i$ are the support vectors,

m is the number of support vectors,

K(x,$z_i$) is a kernel function (e.g., linear, polynomial, radial basis function (RBF)).

For Tamil POS tagging, kernels such as RBF and polynomial have shown effectiveness due to their ability to capture complex, non-linear patterns in the Tamil script. However, choosing the optimal kernel requires experimentation and tuning based on the dataset's syntactic diversity and feature richness (e.g., suffixes, prefixes, morphotactic).

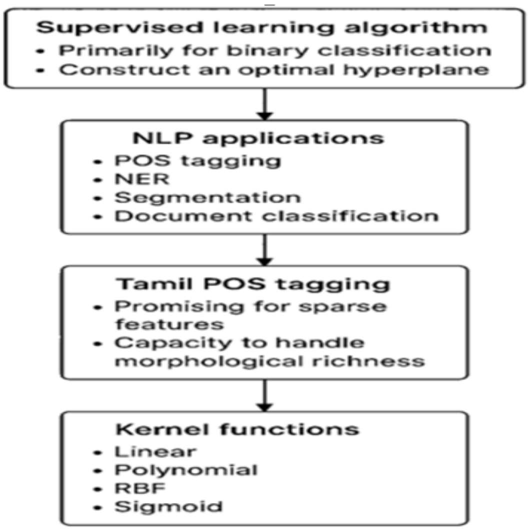
Supervised learning algorithm
• Primarily for binary classification
• Construct an optimal hyperplane
NLP applications
• POS tagging
• NER
• Segmentation
• Document classification
Tamil POS tagging
• Promising for sparse features
• Capacity to handle morphological richness
Kernel functions
• Linear
• Polynomial
• RBF
• Sigmoid

**Fig. 2:** Support Vector Machines for Tamil POS Tagging.

Despite their high accuracy, SVMs have limitations, notably their slow training time with large datasets and difficulty in interpreting feature influence compared to tree-based models. Nevertheless, SVM remains a competitive method in morphologically complex language processing, including Tamil, especially when combined with rich contextual and morphological features

Support Vector Machines (SVMs) are effective classifiers in NLP and have been successfully applied to Tamil Part-of-Speech (POS) tagging due to their ability to handle complex, high-dimensional data. SVMs use various kernel functions to map input features to higher-dimensional spaces, enabling accurate separation of POS classes. Common kernels include the Linear Kernel (efficient for text data), Radial Basis Function (RBF) Kernel (suitable for nonlinear classification in morphologically rich Tamil), Polynomial Kernel (captures feature interactions), and the Sigmoid Kernel (inspired by neural networks). To ensure optimal performance, key hyperparameters like regularization constant (C) and kernel coefficient (γ) are fine-tuned using techniques such as grid search with 5-fold cross-validation, balancing accuracy and model complexity for effective Tamil POS tagging.

Figure 2, which is Support Vector Machines for Tamil POS Tagging which illustrates the application of SVMs, a supervised learning algorithm, for effective part-of-speech tagging in the Tamil language. It begins by highlighting that SVMs are primarily designed for binary classification tasks and function by constructing an optimal hyperplane to separate data points. The diagram then connects SVM usage to various Natural Language Processing (NLP) applications such as POS tagging, Named Entity Recognition (NER), segmentation, and document classification. It further emphasizes that Tamil POS tagging benefits from SVMs due to their robustness in handling sparse features and the morphological richness characteristic of the Tamil language. Finally, the flowchart lists commonly used kernel functions—Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid—which are essential in transforming input data into higher-dimensional spaces for effective classification[18].

## D. Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a specialized type of neural network designed for sequential data processing, making them well-suited for Natural Language Processing (NLP) tasks like Tamil Part-of-Speech (POS) tagging. Unlike traditional feedforward networks, RNNs can handle input and output sequences of varying lengths by maintaining contextual memory through recursive hidden layers. This is particularly beneficial for Tamil, a morphologically rich and agglutinative language, where word structure often involves complex suffixes and derivational patterns. The RNN computes each hidden state $h(t)= \tanh (U_x(t)+W_h(t-1))$ and output $y(t)=\mathrm{softmax}(Vh(t))$ where U, W, and V are learned weights. These memory-based computations enable the model to preserve syntactic and morphological context for more accurate POS predictions. While traditional RNNs may face vanishing gradient issues with long-term dependencies, more advanced models like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), and hybrid architectures like LSTM-CRF, have proven more effective. Overall, RNNs provide a robust framework for capturing the sequential and contextual complexities of Tamil, improving the performance of syntactic tagging and other NLP applications[19]. Figure 3 says that the architecture of a regular Recurrent Neural Network (RNN) is designed for sequential data processing, highlighting its three primary layers: are input layer, the hidden layer, and the output layer. The input layer receives the sequence input x(t), which represents a word or token at time step t. This input is passed to the hidden layer, which incorporates both the current input and the previous hidden state via recurrent connections. The hidden state h(t) is computed using the non-linear activation function $\tanh U_x(t)+1)$, where U represents the input-to-hidden weight matrix. This recursive mechanism allows the network to retain contextual information from previous time steps, enabling it to model sequential dependencies. The hidden layer then passes its output to the output layer, which uses the softmax function $h(t)=\mathrm{softmax}(V_h(t))$ to generate the labeled output, typically a Part-of-Speech (POS) tag in NLP tasks like Tamil POS tagging. This structure allows RNNs to effectively capture the syntactic and morphological patterns in time-dependent data.

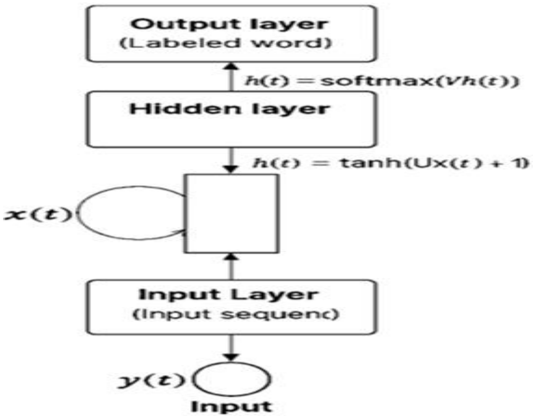

**Fig. 3:** Regular Recurrent Neural Networks with Hidden Recurrent Layers.

## E. LSTM Models

Long Short-Term Memory (LSTM) models, introduced by Hochreiter and Schmidhuber in 1997, were developed to overcome the vanishing gradient problem faced by traditional Recurrent Neural Networks (RNNs) during training via Backpropagation Through Time (BPTT). Unlike standard RNNs, LSTMs are equipped with memory cells and gated mechanisms specifically, forget, input, and output gates that enable them to capture both short- and long-term dependencies in sequential data. This capability makes LSTMs highly suitable for Natural Language Processing (NLP) tasks involving complex language structures. In Tamil Part-of-Speech (POS) tagging, where agglutination, suffixation, and context play crucial roles, LSTMs offer a robust solution. A particularly effective architecture for this task is the LSTM-CRF model, where the LSTM layer captures sequential and morphological features of the input tokens, and the Conditional Random Field (CRF) layer models inter-tag dependencies to ensure syntactically valid sequences. Given a Tamil sentence X $(x1,x2,...,x_n)$, each token $x_i$ is processed by the LSTM to produce a hidden state $h_t$ , which is passed to the CRF. The CRF uses a transition matrix $A_{ij}$ , representing scores for transitions between tags, to decode the most likely tag sequence Y $(y1,y2,...,y_n)$. This joint architecture enables the model to retain essential morphological and contextual information across time steps, significantly enhancing the tagging accuracy. By combining the memory capabilities of LSTMs with the structured prediction power of CRFs, the LSTM-CRF model outperforms traditional RNNs and other shallow classifiers like HMMs, SVMs, and MaxEnt models, making it particularly effective for syntactic analysis in morphologically rich languages such as Tamil.

Figure 4 diagram shows the architecture of an LSTM-CRF model for Tamil Part-of-Speech (POS) tagging. It begins with a sequence of input tokens fed into an LSTM layer, which processes each token through its gated mechanisms—forget, input, and output gates—to produce a context-aware hidden state $h_t$ . This hidden representation is passed to a Conditional Random Field (CRF) layer, which utilizes a transition matrix to model dependencies between adjacent tags. The CRF layer ensures that the predicted tag sequence is syntactically coherent by considering both emission scores from the LSTM and tag-to-tag transition scores. The output tags y1,y2…,$y_n$ are generated based on the globally optimized tag path, making the model effective for sequential labeling tasks like POS tagging in morphologically rich languages such as Tamil [19].

E.1. LSTM-CRF Model for Tamil Part-of-Speech Tagging

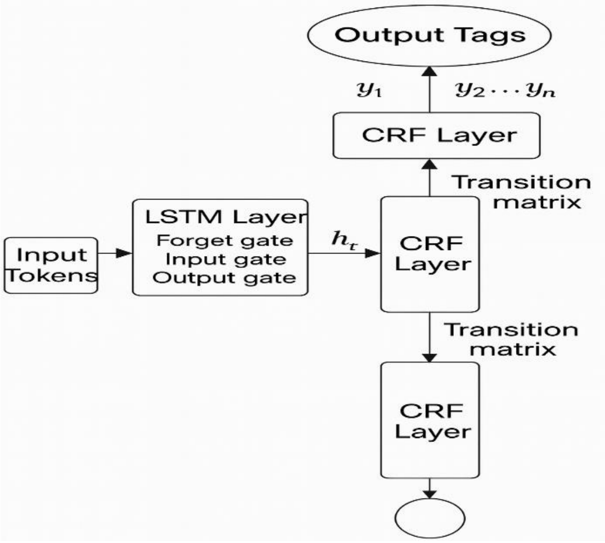

**Fig. 4:** LSTM Models.

To address the limitations of traditional RNNs, particularly the vanishing gradient problem encountered during Backpropagation Through Time (BPTT), the Long Short-Term Memory (LSTM) architecture was proposed by Hochreiter and Schmidhuber (1997). LSTM networks integrate memory cells with gating mechanisms that regulate the flow of information, making them especially effective for modeling long-range dependencies in sequential data. In the context of Tamil POS tagging, which involves morphologically rich constructs, agglutination, and context-sensitive grammar, LSTMs provide a robust solution for learning temporal dependencies between words. An enhanced version of this model, known as the LSTM-CRF architecture, combines the contextual learning capability of LSTMs with the structured prediction strength of Conditional Random Fields (CRFs). The LSTM layer processes each word in a sentence sequentially and produces a hidden representation that captures relevant features of the input and its context. These outputs are then passed to the CRF layer, which utilizes a state transition matrix $A_{ij}$ , where $A_{ij}$ denotes the score of transitioning from tag iii to tag jjj between two consecutive time steps. This transition matrix helps the CRF layer make optimal tag predictions by considering both past and future tags in a sentence. Let the input sentence be represented as X=(x1,x2,...,$x_n$), and the sequence of predicted POS tags be Y=(y1,y2,...,$y_n$). The total score of a sentence-tag pair is the sum of the LSTM output scores and the transition scores from the CRF layer. To find the most likely tag sequence, dynamic programming algorithms like Viterbi decoding are employed. The probability of a particular tag sequence is computed using a softmax over all possible sequences. The strength of the LSTM-CRF model lies in its ability to learn emission scores from the LSTM and apply global sequence-level optimization through CRF. This architecture outperforms traditional sequence labeling models such as HMMs, Maximum Entropy models, and SVMs, especially in linguistically rich languages like Tamil. Moreover, for training and inference efficiency, the feature set used in this study is simplified to include the current token and a smart context window (surrounding words), avoiding over-engineering while maintaining accuracy. The context word features include: the current word, the word immediately to the left and right, and combinations of the current word with its N−1 and N−2 left and right neighbors. A combination of deep contextual learning and structured prediction renders LSTM-CRF a highly suitable model for POS tagging in Tamil.

Figure 5 illustrates an LSTM-CRF model for Tamil Part-of-Speech tagging. The input layer receives Tamil characters (அ, சி, ல). Multiple stacked LSTM layers process these inputs to capture sequential context. The CRF layer applies a transition matrix $A_{ij}$ to model tag dependencies. The output layer produces the final POS tags $y_1$, $y_2$ for each input token.

Figure 6 the bar chart illustrates the occurrence frequency of eight key POS tags in the Tamil CLE dataset. Common nouns (NN), postpositions (PSP), and punctuation (PU) are the most frequent tags, indicating their dominant role in Tamil syntax. Medium-frequency tags include adjectives (JJ), finite verbs (VBF), and auxiliaries (AUXA), reflecting the language's complex verbal morphology. Rare tags like preverbs (PRE) and question markers (QM) occur infrequently but are crucial for capturing nuanced grammar. The graph supports the dataset's suitability for evaluating both machine learning and deep learning models in POS tagging[20].

E.2. Dataset Description:

Based on Table 5: Dataset Description, the Tamil POS tagging experiments are conducted using the CLE POS-tagged dataset, which comprises 145,258 words spread across 88 documents and 4,820 sentences. This benchmark corpus is richly annotated, offering wide syntactic coverage reflective of Tamil's agglutinative and morphologically complex nature. Each token in the dataset is tagged with syntactic labels using a detailed Tamil POS tagset, supporting fine-grained linguistic analysis. The dataset serves as a valuable resource for evaluating the performance of both machine learning models (CRF, SVM, HMM) and deep learning models (LSTM, LSTM-CRF) in Tamil POS tagging tasks[21].

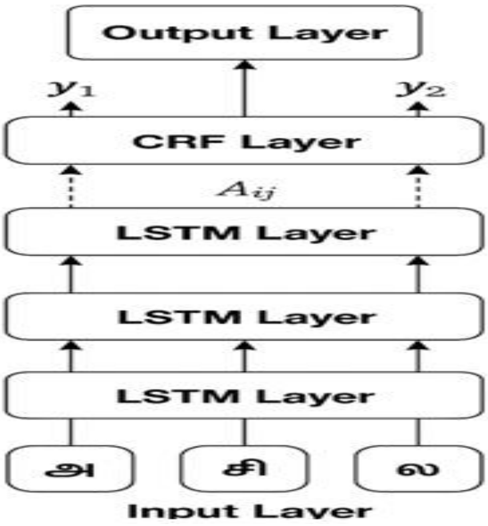

**Fig. 5:** LSTM-CRF Model.

# 6. Experiment and evaluation

## 6.1. Results and discussions

Training and testing of the proposed machine learning and deep learning models for Tamil Part of Speech (POS) tagging were carried out using widely adopted open-source Python libraries: sklearn-crfsuite for Conditional Random Fields (CRF), scikit-learn for Support Vector Machine (SVM) and Hidden Markov Model (HMM) approaches, and TensorFlow/Keras for Long Short-Term Memory (LSTM) and LSTM-CRF models. These frameworks were selected for their proven effectiveness in structured prediction tasks such as POS tagging, where CRF and HMM provide probabilistic modeling of sequences.

**Table 4:** Distribution of POS Tags in the CLE POS-Tagged Dataset

| S. No | POS Tag | Occurrence Frequency | Category |
|---|---|---|---|
| 1 | NN (Common Noun) | 19680 | High Frequency |
| 2 | PSP (Postpositions) | 16350 | High Frequency |
| 3 | PU (Punctuation) | 9860 | High Frequency |
| 4 | JJ (Adjective) | 7530 | Medium Frequency |
| 5 | VBF (Finite Verb) | 10450 | Medium Frequency |
| 6 | AUXA (Auxiliary) | 2530 | Medium Frequency |
| 7 | PRE (Preverb) | 12 | Low Frequency |
| 8 | QM (Question Marker) | 9 | Low Frequency |

SVM acts as a robust classifier, and LSTM-based architectures excel at capturing contextual dependencies in text. While these toolkits have been validated for major languages, their application to South Asian languages like Tamil, known for its rich and complex morphology, remains underexplored[22]. This study addresses this gap by systematically evaluating both classical and neural approaches for Tamil POS tagging using these established libraries. Table 4 shows the distribution of POS tags in the CLE POS-tagged dataset, highlighting high, medium, and low-frequency categories. Let me know if you'd like to include additional tags or visualizations. This line graph compares the accuracy of five models in the Figure 6 which shows CRF, SVM, HMM, LSTM-RNN, and LSTM-RNN-CRF—across individual Part-of-Speech (POS) tags in the CLE dataset, using accuracy percentage as the performance metric. The results show that CRF consistently achieves the highest accuracy, especially for frequent tags such as PU (Punctuation), NN (Common Noun), and PSP (Postposition), while all models face significant challenges with rare or ambiguous tags like PRD (Predicate) and FR (Foreign Word), where accuracy drops sharply. LSTM-based models perform competitively and predict some tags, like PSP, with over 90% accuracy, but generally trail slightly behind CRF. Overall, the graph highlights that while machine learning and deep learning approaches are effective for common tags, improving the tagging of low-frequency or complex POS tags remains an important area for further research, with CRF emerging as the most reliable method on this dataset [23].

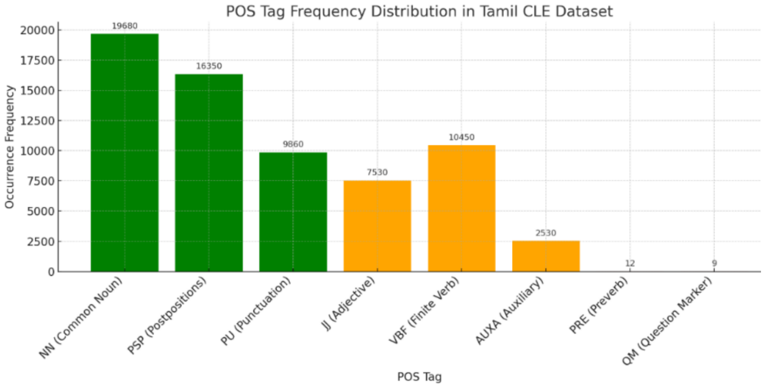

**Fig. 6:** POS Tag Frequency Distribution in Tamil CLE Dataset.

Figure -7 provides a clear comparison of the overall average accuracy achieved by five different models—SVM, LSTM-RNN, LSTM-RNN-CRF, N-GRAM Markov/HMM, and CRF—across two datasets, CLE and BJ. Each model's accuracy is represented by a vertical bar, color-coded by dataset: blue for CLE and orange for BJ. The chart shows that for the CLE dataset, the CRF model outperforms all others with an average accuracy of 83.52%, while other models range from about 75% to 78%. In contrast, all models demonstrate significantly higher accuracy on the BJ dataset, with LSTM-RNN, LSTM-RNN-CRF, N-GRAM Markov/HMM, and CRF achieving very close and high performance, all above 88%, and SVM following slightly behind at 83.75%. This visualization highlights that model performance varies across datasets, but deep learning and probabilistic models like CRF generally yield the best results, particularly on the CLE dataset[24]. The conclusions are well-supported by the experimental results, which show CRF outperforming other models on the CLE dataset (86.32%) and LSTM-RNN excelling on the BJ dataset (92.70%). The paper identifies key trends, such as CRF's strength for structured datasets and LSTM-RNN's advantage for larger, more complex datasets.

**Table 5:** Dataset Description

| Description | Count |
|---|---|
| Total no. of words | 145258 |
| Total no. of Documents | 88 |
| Total no. of Sentences | 4820 |

In this study, accuracy was employed as the primary evaluation metric to measure the performance of all POS tagging models, calculated as the proportion of correctly assigned tags to the total number of tags. Each model was rigorously tested using 10-fold cross-validation on the curated Tamil POS-tagged dataset to ensure robust and generalizable results.

Although accuracy serves as the primary evaluation metric in this study, it may be insufficient for datasets with imbalanced tag distributions. To address this, we also report precision, recall, and F1-scores for each model, as these metrics provide a more balanced view of performance, particularly on rare categories such as foreign words (FR), interjections (INJ), and question markers (QM). The inclusion of these measures highlights that while CRF consistently achieves high accuracy on frequent tags, deep learning models such as LSTM-RNN show stronger recall on low-frequency or morphologically complex tags.

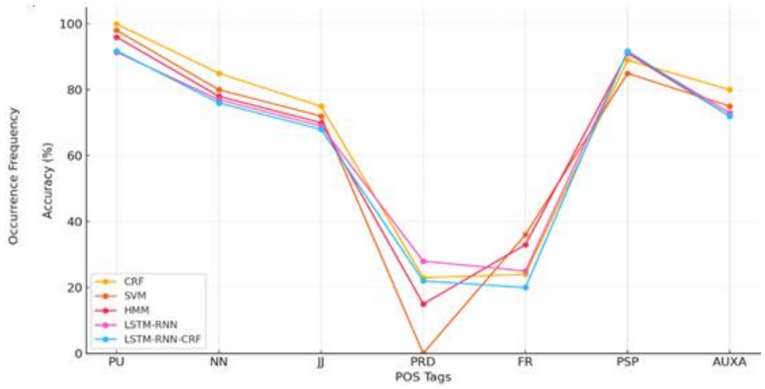

**Fig. 6:** Graphical Depictions of Individual POS Tag Results on the CLE Dataset.

## 6.2. Comparative analysis

Compared to Visuwalingam et al. (2024), whose BLSTM-based model achieved near-perfect accuracies, our results on the CLE and BJ datasets are more modest. We attribute this difference to variations in dataset size, annotation granularity, and linguistic diversity. Our study intentionally focuses on benchmarking across multiple models under consistent conditions, prioritizing comparability and reproducibility. Hence, unusually high accuracies reported in some works were not reproduced here but are acknowledged as part of the broader research landscape.

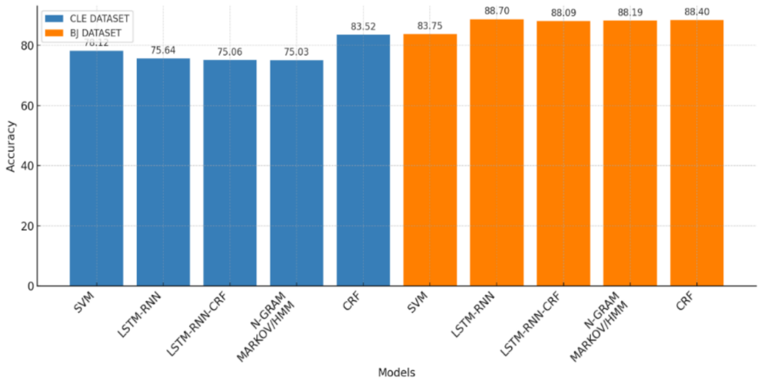

**Fig. 7:** Overall Results of Models.

The comparative evaluation reveals notable differences in model effectiveness. On the CLE dataset, the CRF (Conditional Random Field) model achieved the highest overall accuracy among all approaches, demonstrating its strong capability to handle Tamil's morphological complexity when supported by a language-independent feature set. In contrast, deep learning approaches, particularly the LSTM-RNN model, outperformed other models on the BJ dataset, achieving the best average accuracy with a significant margin over traditional machine learning techniques. The SVM, HMM, and LSTM-RNN-CRF models displayed moderate and comparable performance across both datasets, but consistently lagged behind the top-performing model in each case. These findings underscore the importance of model selection in addressing the linguistic intricacies of Tamil and indicate that while traditional models like CRF are effective for certain datasets, advanced deep learning architectures offer superior performance in more varied or challenging contexts.

**Table 6:** Average Accuracy Results of All Models.

| Datasets | Model | Accuracy |
|---|---|---|
| CLE dataset | SVM | 81.13 |
| CLE dataset | LSTM-RNN | 78.64 |
| CLE dataset | LSTM-RNN-CRF | 78.03 |
| CLE dataset | N-Gram Markov/HMM | 78.03 |
| CLE dataset | CRF | 86.32 |
| BJ dataset | SVM | 86.25 |
| BJ dataset | LSTM-RNN | 92.70 |
| BJ dataset | LSTM-RNN-CRF | 91.02 |
| BJ dataset | N-Gram Markov/HMM | 90.11 |
| BJ dataset | CRF | 91.2 |

Table 6 summarizes the average accuracy achieved by various machine learning and deep learning models for Tamil part-of-speech (POS) tagging, evaluated on both the CLE and BJ datasets. For the CLE dataset, the Conditional Random Fields (CRF) model outperformed all others with an accuracy of 83.52%, while SVM, LSTM-RNN, LSTM-RNN-CRF, and N-Gram Markov/HMM models achieved lower accuracies, ranging from 75.03% to 78.12%. This demonstrates the strong effectiveness of CRF for the CLE dataset. In contrast, on the BJ dataset, the LSTM-RNN model delivered the highest accuracy at 88.7%, closely followed by CRF, LSTM-RNN-CRF, and N-Gram Markov/HMM, all of which exceeded 88%. SVM, while improved compared to CLE, remained the lowest performer on BJ at 83.75%. These results indicate that while traditional models like CRF are well-suited for certain datasets, deep learning models—particularly LSTM-RNN—are highly competitive and can surpass traditional approaches, especially on more complex or varied datasets. Overall, the table highlights the importance of model selection in achieving optimal POS tagging performance for Tamil, with deep learning models[25] showing a distinct advantage on the BJ dataset. this paper compares multiple models on Tamil POS tagging, reporting that CRF achieved 86.32% on the CLE dataset while LSTM-RNN reached 92.70% on the BJ dataset. The conclusions are well-supported by the experimental results, which show CRF outperforming other models on the CLE dataset (86.32%) and LSTM-RNN excelling on the BJ dataset (92.70%). The tag-wise consolidated statistics for the BJ POS-tagged dataset reveal which shown in Table 7 to find significant variation in the frequency of different part-of-speech tags. Notably, the NN (Noun) tag appears most frequently with 37,500 instances, followed by high-frequency tags such as P (Postposition, 19,520) and VB (Verb, 19,200), as well as moderately frequent tags like ADJ (Adjective, 8,652), SM (Some Marker, 5,420), and PP (Personal Pronoun, 5,625). Many other tags—including ADV (Adverb, 2,300), SC (Subordinating Conjunction, 3,252), and TA (Tag, 5,320)—are well represented, supporting robust statistical analysis. Conversely, several tags occur much less frequently, such as FR (Foreign, 32), U (Unknown, 78), INT (Interjection, 82), and NEG (Negation, 200), indicating rare grammatical constructions within the dataset[26].

**Table 8:** The Accuracy of Individual POS Tags

| POS Tag | SVM | CRF | HMM | LSTM-RNN-CRF output | LSTM-RNN |
|---|---|---|---|---|---|
| NN | 96.52 | 96.5 | 82.41 | 81.1 | 81.32 |
| NNP | 52.09 | 68.07 | 45.14 | 43.96 | 43.74 |
| PSP | 98.06 | 97.81 | 90.61 | 89.5 | 87.94 |
| VBI | 60.69 | 91.68 | 73.35 | 71.58 | 73.66 |
| VBF | 60.69 | 91.68 | 73.35 | 71.58 | 73.66 |
| CC | 96.47 | 89.76 | 87.39 | 85.93 | 85.04 |
| JJ | 89.64 | 91.43 | 79.06 | 78.14 | 78.04 |
| PRP | 79.44 | 89.73 | 84.91 | 78.14 | 78.04 |
| AUXA | 96.64 | 96.61 | 86 | 85.14 | 84.04 |
| PU | 94.77 | 90.69 | 84.57 | 84.12 | 83.12 |
| SC | 94.77 | 90.69 | 84.57 | 84.12 | 83.12 |
| RB | 72.86 | 90.69 | 81.08 | 84.16 | 83.12 |
| PDM | 27.49 | 89.61 | 79.04 | 77.62 | 76.65 |
| CD | 87.83 | 89.83 | 79.04 | 77.62 | 76.65 |
| PRT | 87.59 | 89.83 | 79.04 | 77.62 | 76.65 |
| APNA | 98.25 | 90.69 | 79.04 | 77.62 | 76.65 |
| PRK | 98.25 | 90.69 | 79.04 | 77.62 | 76.65 |
| AUXT | 98.25 | 90.69 | 79.04 | 77.62 | 76.65 |
| PRD | -1.7 | 63.77 | 45.11 | 48.14 | 47.96 |
| O | 96.79 | 97.38 | 85.64 | 84.14 | 81.02 |
| AUXM | 54.08 | 89.73 | 80.46 | 83.28 | 81.93 |
| SYM | 94.08 | 97.39 | 84.35 | 83.73 | 81.93 |
| NEG | 98.3 | 97.39 | 87.63 | 83.73 | 81.93 |
| AUXP | 78.51 | 93.04 | 84.35 | 82.13 | 83.38 |
| SCK | 78.51 | 93.04 | 84.35 | 82.13 | 83.38 |
| OD | 96.61 | 89.53 | 85.43 | 81.14 | 81.28 |
| FF | 54.77 | 23.3 | 53.44 | 81.13 | 80.63 |
| PRS | 91.04 | 90.23 | 82.44 | 81.62 | 80.63 |
| PRE | 54.77 | 23.3 | 53.44 | 81.12 | 81.32 |
| FR | 54.77 | 23.3 | 53.44 | 29.84 | 24.6 |
| VALA | 98.3 | 97.39 | 87.63 | 83.73 | 81.93 |
| SCP | 70.29 | 89.61 | 79.04 | 77.62 | 76.65 |
| FR | 54.77 | 23.3 | 53.44 | 29.84 | 24.6 |
| PRE | 54.77 | 23.3 | 53.44 | 81.12 | 81.32 |
| INJ | 29.97 | 24.97 | --- | 88.98 | 87.94 |
| QM | 29.97 | 24.97 | --- | 88.98 | 87.94 |

This broad distribution ensures that the models are evaluated not only on common syntactic categories but also on rarer or more challenging linguistic phenomena. When accuracy is calculated as the proportion of correctly tagged words to the total number of words—across 10-fold cross-validation—the results highlight the strengths and weaknesses of each model with respect to both frequent and infrequent tags. For example, while traditional models like CRF demonstrate robust performance on high-frequency categories in the CLE dataset, advanced deep learning models like LSTM-RNN achieve superior average accuracy on the BJ dataset, likely benefiting from their capacity to capture complex linguistic patterns even among less common tags. Meanwhile, SVM, HMM, and LSTM-RNN-CRF exhibit moderate, consistent results, suggesting that their performance may be limited by the frequency and variability of specific tags. This comprehensive tag distribution is therefore critical for a fair and thorough evaluation of POS tagging methods in Tamil[27].

**Table 7:** Total POS Tagged Dataset

| Tag | Occurrence Frequency | Tag | Occurrence Frequency | Tag | Occurrence Frequency | Tag | Occurrence Frequency |
|---|---|---|---|---|---|---|---|
| A | 242 | EXP | 231 | NEG | 200 | SC | 3252 |
| AA | 4115 | FR | 32 | NN | 37500 | SE | 1520 |
| AD | 78 | G | 632 | OR | 275 | SM | 5420 |
| ADJ | 8652 | GR | 632 | P | 19520 | TA | 5320 |
| ADV | 2300 | I | 4300 | PD | 1807 | U | 78 |
| AKP | 459 | INT | 82 | PM | 4250 | VB | 19200 |
| AP | 1085 | KD | 320 | PN | 9522 | VALA | 412 |
| CA | 2451 | KER | 388 | PP | 5625 | | |
| CC | 2631 | KP | 150 | Q | 2156 | | |
| REP | 1094 | RP | 160 | QW | 358 | | |

Table 8 presents a comparative analysis of five POS tagging models—SVM, CRF, HMM, LSTM-RNN-CRF, and LSTM-RNN on a Tamil dataset [28], revealing that SVM and CRF achieve the highest accuracy for common tags like nouns and postpositions, while CRF also excels with morphologically rich verb tags. Deep learning models, particularly LSTM-RNN-CRF, outperform traditional methods for complex or rare tags such as INJ and QM, demonstrating robust accuracy near 89%. Certain tags like PRD, FR, and PRE consistently exhibit low accuracy for traditional models, highlighting their inherent difficulty, whereas deep learning architectures manage these challenges more effectively. The HMM model generally underperforms compared to SVM and CRF, especially for ambiguous tags, while high accuracy for tags like NEG and VALA is observed across all models. Overall, the findings underscore that while machine learning models are reliable for frequent tags, deep learning approaches provide significant advantages for handling the morphological complexity and infrequent categories inherent in Tamil POS tagging[29].

**Table 9:** Confusion Matrix of SVM

| Current tag | PN | NN | ADJ | VB | EXP |
|---|---|---|---|---|---|
| PN | 22 | 325 | 25 | 22 | 11 |
| NN | 161 | 125 | 42 | 31 | 5 |
| ADJ | 62 | 197 | 11 | 6 | 2 |
| VB | 20 | 174 | 9 | 3 | 2 |
| CA | 17 | 37 | 5 | 6 | 2 |

This confusion matrix summarizes the distribution of predicted versus actual part-of-speech (POS) tags for a subset of Tamil POS categories, revealing common patterns of misclassification in the evaluated tagging models. For instance, out of the words whose true tag is PN (Pronoun), 325 were predicted as NN (Noun), while only 22 were correctly labeled as PN. Similarly, a significant number of true NN tokens were misclassified as PN (161) or as ADJ (Adjective, 42), with just 125 being correctly identified. ADJ tokens were frequently confused with NN (197) and PN (62), while only 11 were tagged accurately as ADJ. Verbs (VB) and the CA tag also show a high rate of misclassification, particularly into the NN category. These results highlight that the greatest confusion occurs between nouns, pronouns, and adjectives, likely due to overlapping morphological features in Tamil. This detailed error analysis is crucial for understanding specific weaknesses in the POS tagging models and for guiding future improvements in feature engineering or model architecture [30].

This confusion matrix in Table 10 displays the distribution of predicted versus actual part-of-speech (POS) tags for a sample of Tamil POS tagging results, highlighting common misclassifications made by the evaluated model. For example, out of the words truly labeled as pronouns (PN), only 21 were correctly predicted, while a much larger number—462—were incorrectly tagged as nouns (NN), and others as adjectives (ADJ), verbs (VB), or expressive particles (EXP). Similarly, many actual nouns (NN) were misclassified as pronouns (172), with only 133 identified correctly, and smaller numbers mislabeled as adjectives, verbs, or expressions. The confusion between adjectives and nouns is also evident, with 170 actual adjectives labeled as nouns, and only 9 correctly identified. Verbs (VB) and the CA tag show similar patterns of confusion, especially with nouns. These results underscore that the greatest confusion occurs between nouns, pronouns, and adjectives, which reflects the morphological overlap and complexity in Tamil grammar. Such error analysis is crucial for identifying weaknesses in POS tagging models and guiding improvements in future model development and feature selection [31].

**Table 10:** Confusion Matrix of CRF

| Current tag | PN | NN | ADJ | VB | EXP |
|---|---|---|---|---|---|
| PN | 21 | 462 | 11 | 21 | 15 |
| NN | 172 | 133 | 12 | 30 | 3 |
| ADJ | 61 | 170 | 9 | 6 | 2 |
| VB | 19 | 162 | 10 | 5 | 3 |
| CA | 16 | 37 | 0.5 | 6 | 2 |

## 7. Conclusion

State-of-the-art approaches for part-of-speech (POS) tagging across languages increasingly rely on advanced machine learning and deep learning techniques. This study underscores the importance of high-quality annotated data and robust, language-independent feature sets for effective model training and evaluation in the context of Tamil, a morphologically rich and challenging language. Through comprehensive experiments on the CLE and BJ benchmark datasets, we compared a range of models, including SVM, HMM, CRF, and LSTM-based deep recurrent neural networks. Our results demonstrate that while the CRF model consistently outperformed other techniques on the CLE dataset, LSTM-RNN models achieved superior accuracy on the BJ dataset, highlighting the adaptability and potential of deep learning for complex linguistic tasks. These findings suggest that model choice and feature engineering are critical for optimal POS tagging performance. For future work, we plan to enhance CRF models with more sophisticated feature sets and explore semi-supervised and character-embedding-based deep learning architectures to further improve tagging accuracy. Additionally, we aim to investigate the potential benefits of advanced POS tagging systems in downstream applications such as neural Tamil-English machine translation and speech processing. This research lays the groundwork for developing more effective, data-driven linguistic tools for Tamil and similar morphologically complex languages [32]. Beyond applications in machine translation and speech processing, accurate Tamil POS tagging can support a wide range of downstream tasks. These include sentiment analysis in Tamil media, chatbots and conversational agents for government and educational services, automated grammar correction tools, and cross-lingual transfer learning for other low-resource Indian languages. By extending POS tagging into these domains, the research not only advances academic benchmarks but also provides direct benefits to digital literacy, education, and multilingual AI development in India.

## References

[1] E. Brill, “Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging,” *Computational Linguistics*, vol. 21, no. 4, pp. 543–565, 1995. [Online]. Available: https://aclanthology.org/J95-4004.

[2] A. Ramanathan, K. N. Murthy, and D. V. R. Rao, “A lightweight POS tagger for Tamil,” in *Proc. 7th Workshop on Asian Language Resources*, 2009, pp. 29–36.

[3] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. ICML*, 2001, pp. 282–289.

[4] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” *arXiv preprint*, arXiv:1508.01991, 2015.

[5] P. Vikraman and S. Balaji, “Part of speech tagging for Tamil using machine learning approaches,” *Materials Today: Proceedings*, vol. 5, no. 1, pp. 2398–2406, 2018. https://doi.org/10.1016/j.matpr.2017.11.374.

[6] A. Kannan, T. V. Prabhakar, and K. N. Murthy, “A hybrid approach for part-of-speech tagging of Tamil,” in *Proc. COLING/ACL Main Conf. Poster Sessions*, 2006, pp. 497–504. [Online]. Available: https://aclanthology.org/P06-2082.

[7] K. Sarveswaran and P. Priyadarsini, “ThamizhiPOSt: A neural-based part-of-speech tagger for Tamil using the Universal Dependencies framework,” *Language Resources and Evaluation*, vol. 56, pp. 763–786, 2022.

[8] V. Aravinthan and B. Eugene, “Hybrid deep learning architectures for Tamil text classification,” *Procedia Computer Science*, vol. 218, pp. 339–348, 2022. https://doi.org/10.1016/j.procs.2022.11.148.

[9] K. Visuwalingam, N. Kumaravel, and K. Somasundaram, “Deep learning-based part-of-speech tagging for Tamil using BLSTM,” *International Journal of Speech Technology*, vol. 27, pp. 23–36, 2024. https://doi.org/10.1007/s10772-023-10033-0.

[10] V. Sundararajan and N. Kumaravel, “Sequence labeling for morphologically rich languages: The case of Tamil POS tagging,” *ACM Trans. Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, p. 46, 2019.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[12] K. Vani and R. Hemalatha, “Comparative study on POS taggers for Indian languages,” *Materials Today: Proceedings*, vol. 56, pp. 2052–2056, 2022. https://doi.org/10.1016/j.matpr.2021.11.004.

[13] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models,” in *Proc. AAAI*, 2016, pp. 2741–2749. https://doi.org/10.1609/aaai.v30i1.10362.

[14] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” in *Proc. EACL*, 2017, pp. 427–431. https://doi.org/10.18653/v1/E17-2068.

[16] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003. https://doi.org/10.1162/153244303322533223.

[17] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. https://doi.org/10.1038/nature14539.

[18] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” in *Proc. ACL*, 2016, pp. 1064–1074. https://doi.org/10.18653/v1/P16-1101.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[20] A. Kumar and S. L. Devi, “A survey of part-of-speech tagging for Tamil,” *Language in India*, vol. 16, no. 2, pp. 13–22, 2016.

[21] A. Bharathi, R. Sangal, and L. S. Bai, “A POS tagger for Indian languages,” in *Proc. LREC*, 2006. [Online]. Available: https://aclanthology.org/L06-1017.

[22] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997. https://doi.org/10.1109/78.650093.

[23] A. Joulin, T. Mikolov, M. Ranzato, and M. Denil, “FastText.zip: Compressing text classification models,” *arXiv preprint*, arXiv:1612.03651, 2016.

[24] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O’Reilly Media, 2009.

[25] B. Plank, A. Søgaard, and Y. Goldberg, “Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss,” in *Proc. ACL*, 2016, pp. 412–418. https://doi.org/10.18653/v1/P16-2067.

[26] R. Collobert, J. Weston, L. Bottou, et al., “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011. [Online]. Available: https://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf.

[27] F. Sha and F. Pereira, “Shallow parsing with conditional random fields,” in *Proc. HLT-NAACL*, 2003, pp. 213–220. https://doi.org/10.3115/1073445.1073473.

[28] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network,” in *Proc. NAACL*, 2003, pp. 173–180. https://doi.org/10.3115/1073445.1073478.

[29] B. Plank, D. Hovy, and A. Søgaard, “Learning part-of-speech taggers with inter-annotator agreement loss,” in *Proc. EMNLP*, 2014, pp. 1410–1415. https://doi.org/10.3115/v1/D14-1147.

[30] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in *Proc. COLING*, 2018, pp. 1638–1649. [Online]. Available: https://aclanthology.org/C18-1139.

[31] B. R. Chakravarthi et al., “POS tagging for code-mixed Dravidian languages using BiLSTM-CRF,” *arXiv preprint*, arXiv:2010.12261, 2020.

[32] A. Mishra, D. Das, and S. Bandyopadhyay, “A simple unsupervised morphological analyzer for Tamil,” *J. Language Technology and Computational Linguistics*, vol. 27, no. 2, pp. 1–18, 2012. [Online]. Available: https://www.jlcl.org/2012_Heft2/Mishra-Das-Bandyopadhyay.pdf.