I.J. Information Technology and Computer Science, **2026, 3, 13-27**
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2026.03.02

Part-of-speech Tagging for Marathi using Maximum Entropy Markov Model

Swati Prakash Sonawane*
School of Computer Science KBC North Maharashtra University, Jalgaon, Maharashtra 425001, India
E-mail: sonawaneswati22@gmail.com
ORCID iD: https://orcid.org/0009-0002-1574-4179
*Corresponding author

Kavita Tukaram Patil
SVKM's Institute of Technology, Dhule, Maharashtra 424001, India
E-mail: meetpatilkavita@gmail.com
ORCID iD: https://orcid.org/000-0002-2893-3270

R. P. Bhavsar
School of Computer Science KBC North Maharashtra University, Jalgaon, Maharashtra 425001, India
E-mail: rpbhavsar@nmu.ac.in
ORCID iD: https://orcid.org/0000-0003-4759-0303

B. V. Pawar
K. C. E. Society's Institute of Management and Research, Jalgaon, Maharashtra 425001, India
E-mail: director@imr.ac.in
ORCID iD: https://orcid.org/0009-0007-2548-1145

Received: 16 December 2025; Revised: 01 February 2026; Accepted: 10 April 2026; Published: 08 June 2026

Abstract: Part-of-Speech (POS) tagging is an essential and important pre-processing activity for many Natural Language Processing (NLP) applications; this is particularly evident for morphologically rich languages such as Marathi. This research investigates POS tagging for Marathi using the Maximum Entropy Markov Model (MEMM). MEMM combines the strengths of conditional probability modelling and sequence prediction, allowing the integration of rich contextual features. Features used include word forms, suffixes, prefixes, and neighboring tags, effectively tackling the challenges presented by inflectional variations and ambiguity in Marathi. Experimental results demonstrate that the MEMM-based POS tagger achieves an accuracy of 83.72%. This performance marks a notable advancement in Marathi POS tagging, given the linguistic diversity and the scarcity of annotated data. Error analysis enhances the issues like ambiguity in homonyms and out-of-vocabulary words, providing methods for further improvement through enriched datasets and sophisticated modelling techniques. This study enhances NLP applications such as machine translation, spell checking, and sentiment analysis for Indian languages and offers a solid foundation for future research in Marathi POS tagging.

Index Terms: Bureau of Indian Standard Tagset, Maximum Entropy Markov Model, Morphologically Rich Languages, Natural Language Processing, Part of Speech Tagging.

1. Introduction

Natural language processing (NLP) is a prominent domain of artificial intelligence which enables computers to analyze and understand human languages. It was formulated to build systems that understand, process and generate natural languages. All NLP systems include pre-processing, main processing, and post-processing phases. Part of Speech tagging is one of the crucial tasks under the preprocessing phase. POS tagging is the process of listing all possible tags for each word of the input sentence and assigning an appropriate tag sequence. A lot of research has already been carried out for resource-rich languages like English, and other Indian Languages. But, research on the Marathi language remained unexplored due to limited resources, morphological complexity, and linguistic diversity [1, 2]. Also, the free word order and highly inflectional features present additional challenges in the development of a POS tagger for Marathi. Various approaches have been proposed for the development of the POS tagger. Here, the MEMM approach has been implemented

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

which uses the BIS tagset to develop a robust POS tagger for Marathi.

The Maximum Entropy Markov Model (MEMM) has the ability to integrate rich contextual features and model sequence dependencies. The BIS tagset, a multilingual tagset, provides a comprehensive categorization of syntactic elements, making it a suitable choice for tagging. An accurate POS tagger for Marathi has far-reaching implications for multiple domains like Sentiment Analysis, Spell Checking [3], Machine Translation and Grammar Correction, Information Retrieval, Speech Processing.

1.1. Motivation and Problem Statement

The lack of high-quality annotated datasets and efficient POS tagging models for Marathi has significantly impeded the development of NLP tools for this language. Ambiguity caused by homonyms, inflectional variations, and borrowed words complicates the tagging process. The motivation behind this research is to bridge this gap by proposing an accurate POS tagging system for Marathi Language.

1.2. Research Highlights

The proposed research highlights are bulleted as follows:

- Innovative Model: Developed a Part-of-Speech (POS) tagging system for Marathi using the Maximum Entropy Markov Model (MEMM). Integrated features such as word forms, suffixes, prefixes, and contextual tags to address Marathi's linguistic challenges.
- Significant Performance: Achieved an accuracy of 83.72%, a notable achievement for POS tagging in a low-resource language like Marathi.
- Comprehensive Dataset: Utilized a 3 lakh word annotated dataset of tourism and health care domain.
- Rich Contextual Features: Employed the BIS tagset for detailed and standardized tagging of syntactic categories in Indian languages.
- Wide Applicability: Demonstrated potential for applications in information retrieval, sentiment analysis, spell-checking, grammar correction, and machine translation.
- Addressing Key Challenges: Tackled inflectional variations, homonyms, and out-of-vocabulary (OOV) words, highlighting areas for improvement in dataset enrichment.

The rest of the paper is organized as follows: Section 2. illustrates the Literature survey to address the work done in this area. The methodology of the proposed work is described in section 3. Section 4 discusses the empirical results obtained by the proposed method and Section 5 describes the conclusion and future scope of the proposed research work.

2. Related Works

Part-of-Speech (POS) tagging has been extensively explored for resource-rich languages, but research on Indian languages, including Marathi, remains limited. Marathi's rich morphological structure and flexible syntax create distinct challenges that demand specialized approaches. This section reviews key studies on POS tagging, with a focus on Marathi and other Indian languages, highlighting the models and datasets used in these efforts.

2.1. POS Tagging for Indian Languages

A lot of work on POS tagging has been carried out for various Indian languages. The languages namely Hindi, Marathi, and Tamil, have similar structure, such as free word order and rich morphology. The prehistoric research for Indian languages was basically done using a rule-based approach. Bharati et al. (1995) advanced a communal framework for POS tagging using linguistic rules for Hindi, later same attuned for other Indian languages. Still, rule based approaches writhed with ambiguous cases and non-dictionary words [4].

Statistical models like Hidden Markov Models (HMMs) announced probabilistic insightful to POS tagging. Rajeev et al. (2008) worked using HMM for Hindi, accomplishing modest precision but fronting challenges in handling non-words and contextual gradations [5]. Conditional Random Fields (CRFs) [10] further improved accuracy by incorporating rich contextual features, as demonstrated by Shrivastava and Bhattacharyya (2008) in their Hindi POS tagging system [6]

2.2. Development of POS Taggers for Marathi

The efforts for the development of a POS tagger for the Marathi language are usually scarce. Deshpande et al. [7] proposed a hybrid approach to get higher accuracy. Singh et al. [8] developed a statistical approach by using unigram, bigram, trigram and HMM model. Deshmukh et al. [9] implemented a deep learning approach on a large dataset.

2.3. Dataset Challenges for Marathi

The noteworthy challenge in POS tagging for Marathi is the lack of across-the-board annotated datasets. The ILMT project backed foundational datasets for Indian languages, containing Marathi, but these datasets were inadequate in size and domain multiplicity. Efforts like the creation of the 3 lakh-word annotated dataset in this study aim to address these gaps and provide a resource for robust model training and evaluation.

2.4. MEMM in POS Tagging

The work on the MEMM has been addressed in the section. The MEMM prepared hybrid using Morkov model and maximum entropy. It has integrated the feature sequence prediction capabilities and flexibility of Markov models and maximum entropy model respectively.

Lafferty et al. (2001) proclaimed MEMMs for sequence tagging tasks, representing its success in controlling sparse data over feature-rich exemplifications [10

Cahyani and Mustikaningtyas[11] recommended the MEMM as an innovative solution for tackling word ambiguity in an Indonesian dataset. The authors have used 10 thousand annotated sentences for POS tagging which applied entropy model to calculate the weights and govern the plausible tag using MEMM bigram and MEMM trigrams. The MEMM bigram is used for considering the sequence of two consecutive words while the MEMM trigram is a three-word sequence. The study demonstrates that the MEMM approach outperforms previous methods, achieving accuracies of 83.04% for the Bigram model and 86.66% for the Trigram model, with the latter showing superior performance due to its lower perplexity values, reflecting a better ability to model word sequences. The authors recommend exploring deep learning methods and developing a maximum entropy function to further improve tag distribution in future research.

Yuwana, Yuliani and Pardede[12] evaluate six widely used methods for POS tagging for the Indonesian language, Unigram, TnT, Brill's Tagger, Naive Bayes, Hidden Markov Model (HMM) and Maximum Entropy Tagger. Their findings reveal that the Maximum Entropy method achieves the highest accuracy, consistently outperforming the other approaches, even with variations in training data size. While the study identifies promising directions for advancing POS tagging, such as using unsupervised pre-training techniques, clustering methods, or deep learning technologies, it notes that the small dataset size limits the overall effectiveness of these methods.

Le-Hong, Roussanaly, Nguyen, and Rossignol [13] explore the maximum entropy approach for Vietnamese text. By incorporating language-specific information, the developed tagger achieves 93.40% overall accuracy and 80.69% accuracy for unknown words on a Vietnamese treebank test set, marking the best results ever reported for Vietnamese POS tagging. The study suggests that this method could be extended to other analytic languages in mainland South Asia. Additionally, the integration of bidirectional dependency networks, local maximum entropy models, and external lexical resources is identified as a way to enhance tagging performance. Future work includes incorporating a comprehensive Vietnamese lexicon to further improve accuracy.

Huang and Zhang [14] highlight the advantages of Maximum Entropy (ME) conditional models over HMM for POS tagging, emphasizing their flexibility in utilizing arbitrary, non-independent features. This capability enables ME models to effectively address sparse training data issues. Experimental results show that ME models reduce POS tagging error rates by 54.25% and 40.56% compared to HMM-based baselines, achieving 98.01% accuracy in close tests and 95.56% accuracy in open tests. The authors recommend future research on feature template design and feature selection algorithms to further enhance both accuracy and computational efficiency.

Pisceldo, Adriani, and Manurung [15] explore the development of Part-of-Speech (POS) tagging for Bahasa Indonesia using probabilistic approaches. The study compares the performance of Maximum Entropy and CRF methods across two distinct corpora. Results indicate that the Maximum Entropy method delivers the highest accuracy. In the future, the research will be extended using large corpora and alternative methods to increase the performance of the POS tagger.

Ekbal, Haque, and Bandyopadhyay [16] developed a Part-of-Speech (POS) tagger for Bengali using the Maximum Entropy model, which preserves contextual information and assorted features to categorize the POS tags. The model was trained using a 72,341-word dataset and exploited a 26-language Indian tagset. The model was tested using 20,000 words, which accomplished an accuracy of about 88.2%. The approach is integrated with word-level features, named entity recognition, and lexicon to enhance the accuracy. This model improved accuracy by about 8% as compared to the HMM-based tagger.

Dandapat [17] presented research on Part-of-Speech (POS) tagging and chunking for various Indian languages as part of the SPSAL shared task contest. It worked on a hybrid approach that uses an ME-based statistical model and a morphological analyzer to upsurge the performance of the tagger. The model accomplished an accuracy of about 88%, 83%, and 68% for Bengali, Hindi, and Telugu, respectively. Furthermore, this research was extended using automatic stochastic tagging of natural language text to outperform the naïve baseline model.

Dalal, Nagaraj, Sawant, and Shelke [18] present a statistical approach for POS tagging and chunking for the Hindi language using the MEMM. The system integrates multiple features, including context-based, word, dictionary, and corpus-based features. The best performance was achieved on the NLPAI-ML 2006 contest corpus, with 88.4% accuracy for POS tagging and 86.45% accuracy for chunking. The authors recommend further error analysis and the inclusion of more language-specific features to enhance system performance, particularly for chunking tasks.

Halácsy et al. [19] evaluate Maximum Entropy (MaxEnt) POS disambiguation systems for Hungarian natural language processing tasks, incorporating an open-source morphological analyzer. The study presents the first application of the maximum entropy framework, which outperforms state-of-the-art tagging methods and effectively handles out-of-vocabulary items, making it suitable for the analysis of large web-based corpora. The proposed system achieves a performance level of 98.17%, comparable to English taggers. The authors suggest that future work will focus on implementing a permissive license system to further enhance accessibility and use.

Huihsin Tseng et al. [20] investigated a Part-of-Speech (POS) tagger for Mandarin Chinese. It emphasized the major

challenge posed by new words compared to the English language. Due to the introduction of a new morphological feature the accuracy improved from 61% to 80%. Moreover, the investigators recommended cross-linguistic similarities between Chinese and German, regardless of their genetic differences. The authors have also suggested noteworthy insights for future POS tagging studies athwart diverse languages.

Toutanova and Manning [21] analyzed a Maximum Entropy-based POS tagger. It includes a wide range of information sources that boosted the performance of the model. It also has the feature wide-range handling of disambiguation of tense forms of verbs, capitalization for unknown words, distinguishing particles from prepositions and adverbs. For experimentation the researcher used the Penn Treebank dataset, which attained 96.86% accuracy for unseen words. However, the authors stated that the full potential of the ME method for POS assignment has yet to be realised.

Adwait Ratnaparkhi [22] devised a statistical model that was used for predicting the POS tags from the annotated corpus. It has the ability to handle the tedious tagging decisions and to put forward the consistency issue of the corpus. The model proposed the strategy which improved the performance. The ME model proposed the flexible linguistic modelling technique which attained the accuracy of 96.6% for unseen test data. The proposed model outperforms the baseline model by 0.5%.

2.5. Gaps and Limitations in Existing Literature

The literature survey depicts the fact that a lot of work has been done on POS tagging for Indian as well as foreign languages. But limited work has been carried out for Marathi language POS tagging. The present POS tagging approaches are grieving by limited dataset size which leads to deprived generalization of non-dictionary words and non-domain words. There is also scarce probing of sequence tagging frameworks like MEMMs and deep learning models. There is a deficit of comprehensive error analysis to discourse the issues posed by morphologically complex and syntactical divergence language like Marathi. Hence, this study aims to fill these gaps by emerging an MEMM-based POS tagger for Marathi, leveraging a large annotated dataset and the BIS tagset to achieve state-of-the-art accuracy.

3. Methodology

The proposed method used comprises various stages as shown in Fig. 1. Each stage effectively contributes to the implementation of POS tagging for Marathi. The first step is the input layer, which consists of input pre-processing, feature extraction, and the feature vector. The process initiates with the pre-processing of the tagged Marathi corpus. The feature extraction phase is responsible for extracting the features and categorizing them into three groups namely word-based, tag-based, and hybrid. For each word in the sentences, an equivalent history tuple is engendered which serves as input to the next stage named Maximum Entropy Markov Model. The MEMM later uses the Viterbi decoding to predict the most likely tag sequences for the words. A detailed explanation of each of the stages is given as follows:

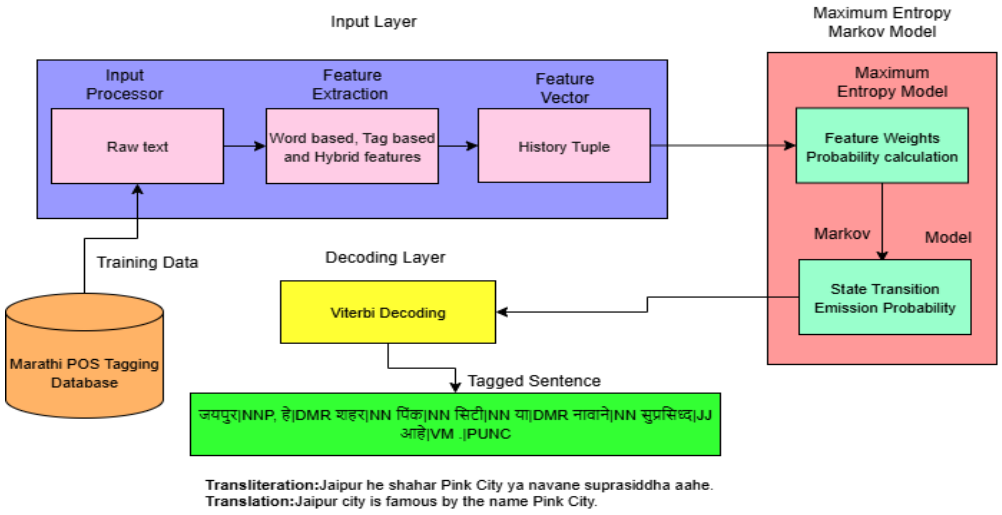


Fig. 1. MEMM POS tagging for marathi process flow

3.1. Dataset

We have used the dataset from [8], which is based on the tourism and health based domain; it consists of 15,145 sentences with 3 lakh annotated words. The dataset was tagged using the BIS tagset, which consists of a comprehensive set of POS categories tailored for Indian languages, capturing linguistic nuances such as morphology and syntactic structures. The training dataset involves the tagged sentences arranged in the format (word1|tagt1, word2|tag2,….,word n|tagn).

3.2. Input Layer

Three blocks make up the input layer: the input processor, feature extraction, and feature vector, which are explained below.

A. Input Processor

The input processor reads the data from the corpus, line by line. It processes each sentence and divides the sentence into a two-dimensional array of words and tags. Each word and tag are split by the "TAGCHAR" which works as a delimiter between word and tag for a sentence. It generates the frequency dictionaries for word, tag, and (word, tag) occurrences in the training corpus, and the data object for the pre-processed data is returned to be managed by feature extraction module.

B. Feature Extraction and Feature Vector

The features are categorized into three sub-categories: word-based features, tag-based features and hybrid features (combination of word-based and tag-based feature). The word based features include prefix, postfix, word length and size of context window of the word. Tag based features include bigram, trigram and occurrences of the tag. Hybrid feature combines the all features of word and tag. The proposed model currently extracts ten features which are in the form of (<t_i, w_(i-n)>,…,<t_i, w_(i-1)>, <t_i, w_i>, <t_i, w_(i+1)>, …, <t_i, w_(i+n)>, where, ti symbolizes tag for current word in the sentence and wi signifies the current word. Each feature in the feature template is represented by feature code starting from (500, …, 510). We use a combination of these features to create a feature vector for each word. The hyper parameter features_cutoff is used to limit the number of features that are considered for each word in the sentence. This is done to reduce the computational complexity of the model, and to prevent overfitting. Whereas, the feature construction and feature cutoff parameters are used to choose the top feature vectors for each word in the given sentence. The model is likely to be overfit if the number of parameters is restricted. The feature vectors are merged to get a single feature vector. The feature vectors are then fed to MEMM model.

C. Maximum Entropy Markov Model

The Maximum Entropy Markov Model (MEMM) is a discriminative model that syndicates forte of maximum entropy models and hidden Markov models. It successfully seizures the complexities of part of speech tagging for the Marathi language. The MEMM model is trained on a large corpus of annotated Marathi text. Every word is allied with a set of features that capture its linguistic properties, such as its uniqueness, word based, tag-based features based on the contextual information. The model uses these features to predict the probability of a tag sequence given a sentence. It uses Maximum entropy model to satisfy the constraints imposed by the training data. The MEMM model is predominantly beneficial for Marathi POS tagging as, it can excellently capture the complexities of Marathi morphology and syntax. It handles the morphological and syntactical features of Marathi very well. It extracts features such as word frequency, tag frequency, and combinations of word and tag frequencies or part of speech tags.

- Probability Calculation:

The probability of each tag given a particular history and features is calculated using the formula given in equation 1:

$$\mathrm{P(tag|history)} = (\mathrm{N/D}) \tag{1}$$

Where, N is the numerator and D refers to the denominator.

The numerator is the dot product of the features vector and the weights vector, and the denominator is the sum of the dot products of all possible tags and the weights vector. The formula is given in equation 2.

$$\mathrm{D} = \sum(\mathrm{d_p(tag_i, W)}) \tag{2}$$

where tag_i is the i-th possible tag, and W is the weights vector.

The features vector is a vector that exemplifies the input features for a specific word in the sentence. The feature vectors are fashioned using feature factory during feature extraction stage. The weight vector characterizes the weight of model for specific tags. These weight vectors are learned during the training stage. The dot product of the features vector and the weights vector is calculated using the following formula given in equation 3:

$$\mathrm{d_p} = \sum(\mathrm{F[i]} * \mathrm{W[i]}) \tag{3}$$

where F[i] is the i-th element of the features vector, and W[i] is the i-th element of the weights vector.

- Gradient Calculation: The gradient is calculated using the following formula given in equation 4:

$$\text{Gradient} = (e_c - o_c) / (\text{D} * \text{regularizer}) \quad (4)$$

where, e_c is expected counts vector, o_c is the observed counts vector, denominator is the denominator for the probability calculation, and regularizer is the regularization parameter.

The expected counts vector is a vector that represents the expected counts for each tag in the sentence. The expected counts vector is calculated using the following formula given in equation 5:

$$e_c[tag_i] = \sum(\text{P(tag_i|history)} * \text{F[i]}) \quad (5)$$

where tag_i is the i-th possible tag, P(tag_i|history) is the probability of the tag given the history, and F[i] is the ith element of the features vector.

The observed counts vector is a vector that represents the observed counts for each tag in the sentence. The observed counts vector is calculated using the following formula given in equation6:

$$o_c[\text{tag_i}] = \sum(1 * \text{features[i]}) \quad (6)$$

where tag_i is the i-th possible tag, and F[i] is the i-th element of the features vector.

- Loss Calculation: The loss method calculates the loss of the model. The loss is calculated using the following formula given in equation7:

$$\text{loss} = -\log(\text{P(tag|history)}) \quad (7)$$

where P(tag|history) is the probability of the correct tag given the history. The loss calculation is used to measure the difference between the model's predictions and the true labels. The goal of training is to minimize the loss.

The following Algorithm 1 shows a detailed description of the MEMM POS Tagging system.

Algorithm1: Marathi POS Tagging Using MEMM

Input:
- Tagged Marathi corpus (⟨word|tag⟩ pairs)
- Predefined feature templates
- Hyperparameters (e.g., features_cutoff, regularization parameter)

Output:
Predicted POS tags for input sentences
Begin:
Preprocessing
Input: Read the tagged corpus line by line.
Process:
Split each sentence into ⟨word, tag⟩ pairs using a delimiter (TAGCHAR).
Populate frequency dictionaries for words, tags, and word-tag pairs.
Output: Preprocessed data in structured format.
Feature Extraction
Input: Preprocessed data and predefined feature templates.
Process:
Generate word-based, tag-based, and hybrid features for each word in a sentence.
Apply feature templates (e.g., ⟨t_i, $w_{(i-1)}$⟩, ⟨t_i, $t_{(i-1)}$, w_i⟩) to derive contextual features.
Use the feature cutoff parameter to select the top relevant features and construct feature vectors for each word.
Output: Feature vectors for all words.
Train MEMM
Input: Feature vectors, training data, and hyper parameters.
Process:
Initialize: Start with random weights for each feature.
Calculate Probabilities:
For each word www and tag ttt:
P(t|history)=d_p(features, weights)∑tagsd_p(features,

$$P(t|history) = \frac{d_p(feature,weights)}{\sum_{tags} d_p(feature,weights)}$$

Gradient Calculation:

$$Gradient = \frac{ee_c - o_c}{D.regularizer}$$

Loss Function: $Loss = -\log(P(t|history))$
Output: Trained weights for MEMM.
Decoding
Input: Trained MEMM model, feature vectors, and test data.
Process:
For each sentence, compute tag probabilities using MEMM.
Apply the Viterbi algorithm for sequence optimization:

- Initialize a trellis with probabilities for the first word's tags.
- For each subsequent word, compute:

$$V(t_i) = max_{ti-1}(V(t_{i-1}).P(ti|t_{i-1}, features))$$

- Backtrack to find the most probable tag sequence.
- Output: Predicted tag sequences.

End

3.3. Decoding Layer

MEMM uses the Viterbi algorithm to predict the most likely sequence of tags that maximizes the probability of the observed feature vector. MEMM optimizes the probability of each tag independently, whereas Viterbi algorithm optimizes the probability of the entire sequence of tags. This means that Viterbi algorithm can capture long-range dependencies and contextual relationships between tags that MEMM may not capture. In many cases, the feature vector may be ambiguous, and MEMM may assign high probabilities to multiple tags.

Viterbi algorithm can handle this ambiguity by selecting the most likely sequence of tags that maximizes the probability of the observed feature vector. Viterbi algorithm is an efficient decoding algorithm that can find the most likely sequence of tags in polynomial time. This makes it suitable for large-scale applications where computational efficiency is important. By using Viterbi algorithm after MEMM, the accuracy of the tag sequence prediction can be enhanced. Viterbi algorithm can correct errors made by MEMM and provide a more accurate sequence of tags. The most likely sequence of hidden states that leads to a sequence of seen events is determined by the Viterbi algorithm. Using the transition and emission probabilities, the method iterates through the series of observed events, computing the probabilities for every hidden state. The algorithm uses various steps to evaluate the probabilities of a given sequence as depicted in Fig.2.

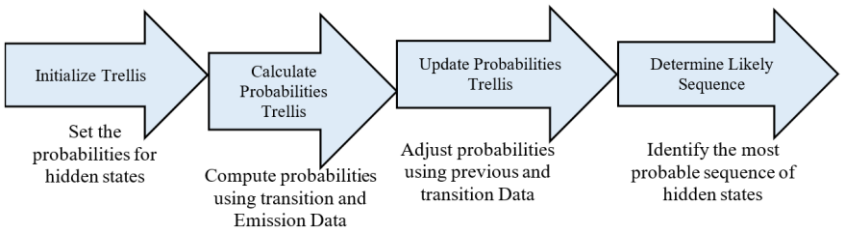


Fig. 2. Working of viterbi algorithm

It initializes the trellis with the probabilities of each hidden state; it is calculated for every state by estimating the transition and emission probabilities. Then, the probabilities of each hidden state at each time step are calculated using the previous probabilities and the transition probabilities. Algorithm uses the trellis to identify the most probable sequence of hidden states that results in the sequence of observed events.

3.4. Output Layer

In this layer the output of the MEMM is fed to the Viterbi algorithm. This layer produces the sequence of tags in order to maximize the probability of the observed feature vector. It produces the sequence of tags, where each tag has a specific class. The sequence of tag can be dispensed to the input feature vector, model's parameters and observed data.

3.5. Mathematical Model for POS Tagging Using MEMM

The problem of part-of-speech (POS) tagging can be mathematically modeled as follows:

A. System Description

Input: A sentence S={w1,w2,…,wn}, where wi is the ith word in the sentence. Output: A sequence of tags T={t1,t2,…,tn},, where ti is the tag assigned to word wi . Objective: Find the most probable sequence of tags T* given the sentence S and the features F given in equation 8:

$$T *= argmax_T P(T|S,F) \tag{8}$$

B. Feature Representation

The feature vector F is constructed for each word wi using:

- Word-based features: Contextual words w_(i-2),w_(i-1),w_i,w_(i+1),w_(i+2).
- Tag-based features: Previous tags t_(i-2),t_(i-1).
- Hybrid features: Combinations of words and tags.

Feature templates ϕ (w_i,t_i) are defined as given in the equation 9:

$$\phi_K (w_i, t_i) = \begin{cases} 1 & if\ feature\ k\ is\ active\ for\ (w_i, t_i) \\ 0 & otherwise \end{cases} \tag{9}$$

The feature vector for a word is given in the equation 10:

$$F_i=[\phi_1 (w_i, t_i), \phi 2\ (w_i, t_i), \dots, \phi k\ (w_i, t_i)] \tag{10}$$

where K is the total number of features.

C. Probability Calculation Using MEMM

The probability of a tag ti for word wi given its history hi (contextual and tag-based features) is given in the equation 11:

$$P(ti|hi) = \frac{\exp\ (\sum_k \lambda_k \phi_k(\text{wi,ti}))}{\sum_{t' \epsilon T} \exp\ (\sum_k \lambda_k \phi_k(\text{wi,ti}))} \tag{11}$$

Where λ_k are the learned weights for feature ϕ_k, T is the set of all possible tags.

D. Sequence Prediction Using Viterbi Algorithm

The best sequence of tags T* is computed using the Viterbi algorithm using formula given in the equation 12:

$$V\ (t_i) = max_{ti-1}(V(t_{i-1}).P(ti|t_{i-1}, features)) \tag{12}$$

where $V\ (t_i)$is the probability of the most likely tag sequence ending with tag t at position i.

E. Model Training

Training involves learning the weight vector λ by maximizing the log-likelihood L given in the equation 13:

$$L(\lambda) = \sum_{i=0}^{N} \log P(T|S,F) \tag{13}$$

where N is the number of sentences in the training data.

The gradient of L(λ) is given in the equation 14:

$$\frac{\partial L}{\partial \lambda_k} = \text{Observed Count} - \text{Expected Count} \tag{14}$$

where:

Observed Count: Frequency of feature ϕ_kin the training data.

Expected Count: Predicted frequency of feature ϕ_k under the model.

Optimization is performed using the L-BFGS algorithm.

The output of the MEMM model: tags assigned to each word according to the BIS Tag Set such as:

Input: दिल्लीमधील जामा मस्जीद ही भारतामधील सर्वात मोठी मशीद आहे, आणि ती शाह जहानने १९५६ मध्ये बांधली.

Transliteration: Dillīmadhīl Jāmā Masjid hī Bhāratāmadhīl sarvāta mōṭhī Mashīd āhe, āṇi tī Shāh Jahānne 1956 madhye bāndhlī.
Translation: The Jama Masjid in Delhi is the largest mosque in India, and it was built by Shah Jahan in 1956.
Output: <दिल्लीमधील:NNP, जामा:NNP, मस्जीद:NN, ही:DMR, भारतामधील:NN, सर्वात:INTF, मोठी:JJ, मशीद:NN, आहे:VM, ,:PUNC, आणि:CCD, ती:DMR, शाह:NNP, जहानने:NNP, १९५६:NN, मध्ये:NST, बांधली:VM, .:PUNC.

4. Results and Discussion

4.1. Results

The proposed Maximum Entropy Markov Model (MEMM) for Part-of-Speech (POS) tagging of Marathi text was evaluated using a dataset of 15,145 sentences, divided into training (70%) and validation (30%) subsets. The model achieved an accuracy of 83.72%, highlighting its capability to effectively handle the complexities of Marathi morphology and syntax. Accuracy is defined as the ratio of the number of correctly predicted tags to the total number of tags in the dataset. The accuracy of the model can be calculated by using the following equation 15:

$$\text{Accuracy}=\frac{\text{Number of Correct Tags(Ct)}}{\text{Total Number of Tag(Nt)}}\times 100 \tag{15}$$

Where,

- Number of Correct Tags (Ct): The count of tags predicted correctly by the MEMM model when compared to the ground truth tags in the validation set.
- Total Number of Tags (Nt): The total number of words (and their corresponding tags) in the validation dataset.
- Accuracy (%): The result is multiplied by 100 to express the value as a percentage.

Example Calculation: If the validation dataset contains 45,000 tags, and the model correctly predicts 35,271 of them, the calculation is given in equation 16:

$$\text{Accuracy}=\frac{37,674}{45,000}\times 100 \approx 83.72\% \tag{16}$$

Thus, for an accuracy of 83.72%, the model must correctly predict 37,674 tags out of 45,000.

The dataset is differentiated into two sections, a training dataset and a testing dataset, with a ratio of 7:3. During preprocessing, the sentences are processed by the Input Processor, which breaks them down into arrays containing individual words paired with their respective tags.The dataset is split into ten folds for k-fold cross validation; each fold has a slightly different accuracy and each fold contains approximately 1500 sentences.

Table 1. K fold cross validation

Fold name	Average accuracy(%)	Maximum accuracy(%)	Minimum accuracy(%)	Median accuracy(%)
fold_0	82.42	100	13	84.21
fold_1	81.74	100	40	83.33
fold_2	82.80	100	33.33	84.61
fold_3	82.59	100	25	83.78
fold_4	83.72	100	44	84.61
fold_5	82.46	100	28.57	833.33
fold_6	83.05	100	37.5	84.61
fold_7	83.41	100	33.33	85
fold_8	83.03	100	33.33	84.61
fold_9	83.66	100	28.57	84.61
Average Accuracy	**83.72%**	**100**	**13%**	**84.61%.**

Table 1 (K-fold Cross Validation) shows the accuracy metrics of each fold with average, maximum, minimum, and median accuracy. The maximum accuracy for all folds consistently reached 100%. The minimum accuracy values, however, displayed significant variation, ranging from 13% (fold_0) to 44% (fold_4). The average accuracy across individual folds falls within a narrow range of approximately 81.74% to 83.72%. Median accuracy values remained close to the fold averages, lying between 83.33% and 85%, which indicates consistent central tendencies in the model's performance. Fig. 3 shows the graphical representation of the accuracy metrics of each fold.

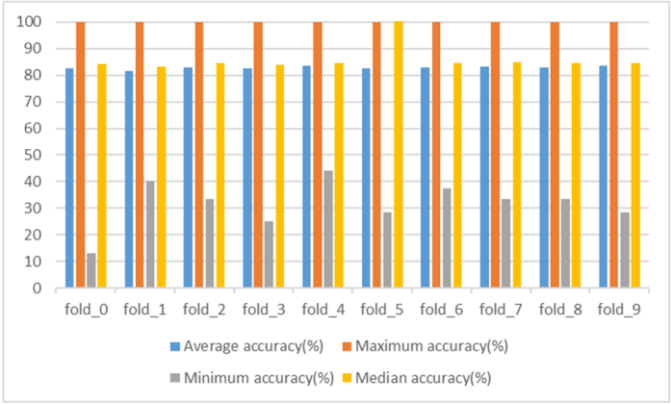


Fig. 3. Accuracy metrics of each fold

To compute Precision, Recall, and F1-score, we need more than just the number of correct predictions and total tags. Specifically, we need:

- True Positives (TP) – Correctly predicted tags.
- False Positives (FP) – Incorrectly predicted tags (predicted but actually incorrect).
- False Negatives (FN) – Tags the model failed to predict (actual tags not predicted).
- The number of times the model accurately predicted the negative class is defined as the True Negative (TN).

Table 2. Confusion metrics

Actual Tags	**Predicted Tags**	
	Predicted Positive	**Predicted Negative**
Actual Positive	TP = 35,271	FN = 9,729
Actual Negative	FP = 4,729	TN = 15,000

- **Precision**

$$Precision = \frac{\text{TP}}{\text{TP+FP}} = \frac{35{,}271}{35{,}271+4{,}729} = \frac{35{,}271}{40{,}000} = 0.8818 \tag{17}$$

$$Precision = \mathbf{88.18}\%$$

- **Recall**

$$Recall = \frac{\text{TP}}{\text{TP+FN}} = \frac{35{,}271}{35{,}271+9{,}729} = \frac{35{,}271}{45{,}000} = 0.7838 \tag{18}$$

$$Recall = \mathbf{78.38}\%$$

- **F1-Score**

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{19}$$

$$F1 - Score = 2 \times \frac{0.8818 \times 0.7838}{0.8818+0.7838} = 2 \times \frac{0.6908}{1.6656} \approx 0.8291$$

$$F1 - Score = \mathbf{82.91}\%$$

Table 3. Calculated performance metrics

Metric	**Value**
Accuracy	83.78%
Precision	88.18%
Recall	78.38%
F1-Score	82.91%

Fig. 4 shows the graphical representation of Performance Metrics.

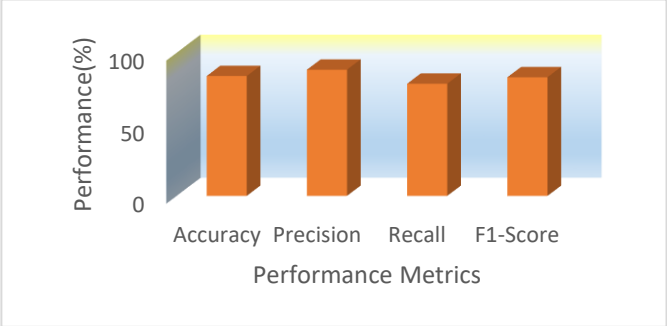


Fig. 4. Graph of performance metrics

4.2. Discussion

A. Feature Effectiveness

The inclusion of word-based, tag-based, and hybrid features in the feature extraction stage proved instrumental in capturing linguistic subtleties unique to Marathi. Features like <ti, wi+1> and <ti, wi-1, wi> captured contextual relationships effectively, aiding in accurate tag predictions.

B. MEMM's Contribution

The MEMM model's discriminative approach, which combines the strengths of maximum entropy models and hidden Markov models, facilitated robust handling of Marathi's inflectional morphology. The model has the ability to distinguish the complex linguistic features due to the use of customized features.

C. Viterbi Algorithm's Role

It considers the tagged sequence that individual tag. It is used for optimizing the global probability of the tagged sequence to heighten the sequence level accuracy. The Viterbi decoding algorithm heightened sequence-level accuracy by optimizing. It has the ability to handle the ambiguous feature vectors which will help to improve the accuracy of the tag sequence.

D. Model Accuracy

Though, the model has secured praiseworthy accuracy, there is scope for perfection especially, in case of Out-of – Vocabulary words, and rare word-tag combinations. While the model achieved praiseworthy accuracy, there is room for perfection, especially in handling rare word-tag combinations and out-of-vocabulary (OOV) words.

E. Comparative Analysis

The attained accuracy lines up well with erstwhile studies on resource-scarce languages, but, morphological complexity often poses a challenge. However, with larger datasets and enhanced feature engineering, the model's accuracy could be further improved.

- Error Analysis

The error analysis of POS categories is shown in Table 4; ambiguity mostly occurred between Main Verb (VM) and Auxiliary Verb (VAUX), as well as between Noun(NN) and Adjective(JJ).

Table 4. Error analysis

Actual Tag	Assigned Tag	Count
VM	VAUX	7517
VAUX	VM	4088
JJ	NN	2771

- Main verb(VM) Vs Auxiliary Verb(VAUX)

In Marathi, auxiliary verbs typically follow a main verb root to indicate tense, aspect, or mood. Depending on the context, various verb forms can function as both main verbs and auxiliaries, thereby adding complexity.

For Example: आहे (ahe).

Input Sentence 1: ऑक्टोबर ते मार्च हा कालावधी जयपुरला भेट देण्यासाठी सर्वोत्तम आहे.

Transliteration: October te March ha kaalavadi Jaipurla bhet denyasathi sarvottam aahe.

Translation: The best time to visit Jaipur is from October to March.

Output POS Tags: ऑक्टोबर|NN ते|DMR मार्च|NN हा|DMR कालावधी|NN जयपुरला|NNP भेट|NN देण्यासाठी|VM सर्वोत्तम|JJ आहे|VM .|PUNC

Input Sentence 2: राजस्थान पर्यटन विकास मंडळाने आर टीडी सी लालगढ राजवाड्याच्या काही भागाचे होटेलमध्ये रुपांतर केले आहे.

Transliteration: Rajasthan Paryatan Vikas Mandḷāne RTDC Lālgaḍh Rajvāḍyāchya kāhī bhāgāche hoṭelmadhe rūpāntar kele āhe.

Translation: Rajasthan Tourism Development Board has converted some parts of RTD C Lalgarh Palace into a hotel.

Output POS Tags: राजस्थान|NNP पर्यटन|NN विकास|NN मंडळाने|NN (|SYM आर|NN टी|NN डी|NN सी|NN)|SYM लालगढ|JJ राजवाड्याच्या|NN काही|QTF भागाचे|NN होटेलमध्ये|NN रुपांतर|NN केले|VM आहे|VAUX .|PUNC

In example sentence 1 and sentence 2, word आहे (ahe) shows the ambiguity between the main verb and the auxiliary verb.

- Noun(NN) Vs Adjective(JJ)

Like many Indian languages, Marathi has a high degree of morphological richness, meaning that depending on the way a word is used in a phrase, it can be both an adjective and a noun.

For example: शांत (shant)

Input Sentence 3: राजस्थानच्या सुंदर शांत व रंगीबेरंगी भूमीचा शोध घेण्याची एक उत्कृष्ट संधी मिळते.

Transliteration: Rajasthanchya sundar shant va rangiberangi bhumicha shodh ghenyachi ek utkṛṣṭ sandhi milate

Translation: It provides an excellent opportunity to explore the beautiful, peaceful and colorful land of Rajasthan.

Output POS Tags: राजस्थानच्या|NNP सुंदर|JJ ,|PUNC शांत|JJ व|CCD रंगीबेरंगी|JJ भूमीचा|NN शोध|NN घेण्याची|VM एक|QTC उत्कृष्ट|JJ संधी|NN मिळते|VM .|PUNC

Input Sentence 4: जवळपास ख्रिस्तपूर्व २०० असलेल्या भाजा गुंफांची वास्तुशास्त्रीय आकर्षकता , भोवतालच्या घनदाट हिरवाईमध्ये असून त्यांना गूढरम्य शांततेचे कोंदण लाभले आहे.

Transliteration: Javalpās Khristapūrv 200 aslelyā Bhājā gufānchī vāstuśāstrīya ākarṣaktā, bhovtālyā ghanadāṭ hirvāīmadhye asūn tyānnā gūḍhramya śānteteche kondan lābhalē āhe.

Translation: The architectural charm of the Bhaja Caves, dating back to around 200 BC, is set amidst the dense greenery that surrounds them, giving them an aura of mystical tranquility.

Output POS Tags: जवळपास|RB ख्रिस्तपूर्व|JJ २००|NN असलेल्या|VM भाजा|NN गुंफांची|NN वास्तुशास्त्रीय|NN आकर्षकता|NN ,|PUNC भोवतालच्या|NNP घनदाट|JJ हिरवाईमध्ये|NN असून|VM त्यांना|PRP गूढरम्य|NN शांततेचे|NN कोंदण|NN लाभले|VM आहे|VAUX .|PUNC

In example sentence 3 and sentence 4, word शांत (shant) shows the ambiguity between the noun and the adjective.

The right POS is frequently determined by the syntactic position and surrounding words, although taggers could incorrectly identify them due to not having sufficient contextual knowledge.

- Out-of-vocabulary OOV words are handled using the most frequent tag strategy, combined with bigram and trigram contexts and surrounding word information.

5. Conclusions and Future Scope

From the result of proposed research, it emanated to know that the MEMM effectively reveals the part-of-speech (POS) tagging for Marathi, with a proficient accuracy of 83.72%. The model successfully captures contextual and morphological nuances of Marathi text by assimilating word-based, tag-based, and hybrid features. The addition of the Viterbi algorithm for decoding enhances the sequence tagging process, which guarantees the robust performance. It reveals the fact that the MEMM framework is well suited for various applications like spell-checking, syntactic parsing, grammar checking for Marathi language.

The future work comprises expansion of annotated corpora which will cover diverse domains. Future work also includes the use of a hybrid model that will combine MEMM with a neural network or deep learning model to improve the performance of the proposed approach. These efforts could ominously progress the NLP for Marathi and could greatly contribute to the broader initiatives for resource-scarce languages.

All the Declarations and Statements

Author Contributions Statement
Swati Prakash Sonawane: Writing, Drafted the initial manuscript, contributed to the literature survey, and documented the technical background of the study, Conceptualization, Methodology, Proposed research ideas, Constructed the overall framework.
Kavita Tukaram Patil: Model Training, Validation, and Performance Evaluation: Led the model training process, validated results using standard metrics, and benchmarked performance against existing methods.
R. P. Bhavsar: Formal Analysis, Visualization, and Statistical Analysis: Performed in-depth analysis of experimental results, and supervised project execution.
B. V. Pawar: supervised the validated results using standard metrics.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement
The authors declare no conflicts of interest.

Funding Declaration
This research was self funded.

Data Availability Statement
The dataset is not publicly available; we have taken this dataset from Prof. Nisheeth Joshi, Department of Computer Science, Banasthali University, Rajasthan, India. Same details of the dataset are published in the paper [8].

Ethical Declarations
The authors should declare the following: Statements of ethical approval for studies involving human subjects and/or animals.

Acknowledgments
We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.
We sincerely thank Prof. Nisheeth Joshi, Department of Computer Science, Banasthali University, Rajasthan, India, for providing the dataset used in this study. We also extend our gratitude to the Centre for Modelling Simulation and Design (CMSD) research laboratory at the University of Hyderabad for facilitating the implementation of this work. Our special thanks to Prof. Rajeev Wankar, School of Computer and Information Sciences, Central University of Hyderabad, for his valuable guidance and support throughout this research.

Declaration of Generative AI in Scholarly Writing
https://www.mecs-press.org/ijcnis/author.html#DeclarationofgenerativeAIinscholarlywriting
Authors should disclose in their article how AI and AI-assisted technologies were used during the writing process.

Abbreviations
The following abbreviations are used in this manuscript:

AI - Artificial Intelligence
NLP - Natural Language Processing
DL - Deep Learning
MEMM- Maximum Entropy Markov Model
POS- Part of Speech

References

[1] K. T. Patil, R. P. Bhavsar, and B. V. Pawar, "Contrastive study of minimum edit distance and cosine similarity measures in the context of word suggestions for misspelled Marathi words," Multimedia Tools and Applications, Vol. 82, No. 10, pp. 15573–15591, 2023, doi: 10.1007/s11042-022-13948-z.
[2] K. T. Patil, R. P. Bhavsar, and B. V. Pawar, "Word suggestions for non-word text errors using similarity measure," in Proceedings of the 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Vol. 1, pp. 892–897, IEEE, 2021. DOI: 10.1109/ICACCS51430.2021.9441858
[3] K. T. Patil, R. P. Bhavsar, and B. V. Pawar, "Spelling checking and error corrector system for Marathi language text using minimum edit distance algorithm," in Advances in Computing and Data Sciences: 5th International Conference, ICACDS 2021,

Revised Selected Papers, Part I, pp. 102–111, Springer International Publishing, 2021, doi: 10.1007/978-3-030-81462-5_10.
[4] A. Bharati, R. Sangal, and V. Chaitanya, Natural Language Processing: A Paninian Perspective, Prentice Hall of India, 1995. (Book - No DOI)
[5] N. Joshi, H. Darbari, and I. Mathur, "HMM based POS tagger for Hindi," in Computer Science & Information Technology (CS & IT): Third International Conference on Computer Science and Information Technology (CCSIT), pp. 341-349, AIRCC, 2013, doi: 10.5121/csit.2013.3639.
[6] M. Shrivastava and P. Bhattacharyya, "Hindi POS tagger using naive stemming: harnessing morphological information without extensive linguistic knowledge," in Proceedings of the ICON-08, Pune, India, 2008. (Conference paper - DOI not assigned)
[7] Deshpande, Madhuri M., and Sharad D. Gore. "A hybrid part-of-speech tagger for Marathi sentences." In 2018 International Conference on Communication information and Computing Technology (ICCICT), pp. 1-10. IEEE, 2018. DOI: 10.1109/ICCICT.2018.8325898
[8] Singh, Jyoti, Nisheeth Joshi, and Iti Mathur. "Development of Marathi part of speech tagger using statistical approach." In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1554-1559. IEEE, 2013. DOI: 10.1109/ICACCI.2013.6637411
[9] R. D. Deshmukh and A. Kiwelekar, "Deep learning techniques for part of speech tagging by natural language processing," in 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, pp. 76-81, IEEE, 2020, doi: 10.1109/ICIMIA48430.2020.9074941.
[10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in Proceedings of the 18th International Conference on Machine Learning (ICML), pp. 282–289, 2001. (Classic foundational paper - predates widespread DOI assignment for ML conferences)
[11] D. E. Cahyani and W. Mustikaningtyas, "Indonesian part of speech tagging using maximum entropy Markov model on Indonesian manually tagged corpus," IAES International Journal of Artificial Intelligence, vol. 11, no. 1, pp. 336-344, 2022. DOI: 10.11591/ijai.v11.i1.pp336-344.
[12] R. S. Yuwana, A. R. Yuliani, and H. F. Pardede, "On part of speech tagger for Indonesian language," in Proc. 2nd Int. Conf. on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 2017, pp. 369-372. DOI: 10.1109/ICITISEE.2017.8285530.
[13] P. Le-Hong, A. Roussanaly, T. M. H. Nguyen, and M. Rossignol, "An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts," in Proceedings of TALN 2010, p. 12, 2010. (Conference paper - DOI not assigned for this regional conference) https://aclanthology.org/2010.jeptalnrecital-long.36.pdf
[14] H. Huang and X. Zhang, "Part-of-speech tagger based on maximum entropy model," in Proc. 2nd IEEE Int. Conf. on Computer Science and Information Technology, Beijing, China, 2009, pp. 26-29. DOI: 10.1109/ICCSIT.2009.5234787.
[15] F. Pisceldo, M. Adriani, and R. Manurung, "Probabilistic part of speech tagging for Bahasa Indonesia," in Proceedings of the Third International MALINDO Workshop, pp. 1–6, 2009. (Workshop paper - DOI not assigned)
[16] A. Ekbal, R. Haque, and S. Bandyopadhyay, "Maximum entropy based Bengali part of speech tagging," Research in Computing Science Journal, Vol. 33, No. 8, pp. 67–78, 2008. (Regional journal - DOI not found in standard databases) http://www.cicling.org/2008/RCS-vol-33/06-Ekbal.pdf
[17] S. Dandapat, "Part of speech tagging and chunking with maximum entropy model," in Proceedings of the IJCAI Workshop on Shallow Parsing for South Asian Languages, pp. 29–32, 2007. (Workshop paper - DOI not assigned)
[18] A. Dalal, K. Nagaraj, U. Sawant, and S. Shelke, "Hindi part-of-speech tagging and chunking: A maximum entropy approach," Proceedings of the NLPAI Machine Learning Competition, 2006. (Competition paper - DOI not assigned)
[19] P. Halácsy, A. Kornai, C. Oravecz, V. Trón, and D. Varga, "Using a morphological analyzer in high precision POS tagging of Hungarian," in Proc. Fifth Int. Conf. on Language Resources and Evaluation (LREC 2006), Genoa, Italy, 2006, pp. 2245-2248. DOI: 10.63317/5ofkqpeogev6.
[20] H. Tseng, D. Jurafsky, and C. D. Manning, "Morphological features help POS tagging of unknown words across language varieties," in Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, 2005. (Workshop paper - DOI not assigned) https://aclanthology.org/I05-3005.pdf
[21] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63–70, 2000. (Conference paper - predates widespread DOI assignment) https://aclanthology.org/W00-1308.pdf
[22] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 1996. (Classic foundational paper - predates DOI assignment for NLP conferences) https://aclanthology.org/W96-0213.pdf

Authors' Profiles



Mrs. Swati Prakash Sonawane received Bachelor of Computer Science in 2012. She completed masters in Computer Science in 2014. She has four years of experience in the teaching profession. Currently she is pursuing Ph.D. (Computer Science) from School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. Her research areas of interest include Natural Language Processing. She presented 3 papers and 1 poster at international conferences and 1 paper in a journal and also attained 2 FDPs.



Dr. Kavita Tukaram Patil working as an Assistant Professor in the Department of Computer Engineering at SVKM's Institute of Technology, Dhule, Maharashtra. She is graduated in Computer Engineering at D. N. Patel College of Engineering, Shahada, Nandurbar, Maharashtra, India. She secured Master of Computer Science and Engineering at SSBT's COET, Bambhori, Jalgaon, Maharashtra, India. She secured Ph.D. in Computer Engineering at School of Computer Sciences, KBCNMU, Jalgaon, Maharashtra, India. Apart from academic qualification she also qualified UGC-NET-2023 in computer science and application. She has organized 13 FDP and attended more than 57 FDPs. She has delivered two expert talk on "Project Guidelines and Research Paper Writing". She works as RPS committee member for Ph.D. Candidates at Sandip University Nashik. She is in teaching profession for more than 13 years. She has presented and published 18 papers in National and International Journals, Conference and Symposiums. She also published one copyright, two patent, one book, one scopus indexed book chapter. She is honored with "Research Excellence Award-2024" by Institute of Scholars. Her main area of interest includes MANET, Artificial Intelligence, Natural Language Processing, Machine Learning, and Deep Learning.



R. P. Bhavsar completed his B.C.S. degree from Pune University in 1992, followed by an MCA in 1995 from Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. He received his Ph.D. in Computer Science from the same university in 2016. He has varied experience of 28 years which includes 03 years as Scientist at C-DAC (Pune), 07 years as System Analyst, 18 years of teaching. Currently, he is working as Professor in School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon. His research focuses on Machine Translation, Applied Natural Language Processing, Capacity Building, Machine Learning, IT and Networking infrastructure setup, and software development projects. Currently, he is supervising the research work of 04 Ph.D. scholars.



B. V. Pawar completed his B.E. in Production Engineering from the renowned VJTI Mumbai in 1986, followed by an M.Sc. in Computer Science from Mumbai University in 1988. He completed his Doctor of Philosophy (Ph.D.) in Computer Science at Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, India. With an extensive background spanning 34 years in teaching and research in computer science. Presently, he is working as Director of K. C. E. Society's Institute of Management and Research, Jalgaon. His research interests encompass NLP, Web computing, Information Retrieval, Machine Translation, and Machine Learning. Dr. Pawar has supervised 17 research scholars in their doctoral research and has a prolific publication record, with over 89 research/journal papers, 128 conference paper presentations, and 05 invited talks.

How to cite this paper

Swati Prakash Sonawane, Kavita Tukaram Patil, R. P. Bhavsar, B. V. Pawar, "Part-of-speech Tagging for Marathi using Maximum Entropy Markov Model", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.3, pp.13-27, 2026. DOI:10.5815/ijitcs.2026.03.02