RESEARCH ARTICLE

Comparison of Machine Learning and Deep Learning Models for Part-of-Speech Tagging

Aftab Ahmad Khan[1], Wahab Khan[1], Muhammad Alamzeb Khan[1], Khairullah Khan[1], Fida Muhammad Khan[2,*], Atta Ur Rahman[3], Hazrat Bilal[4,*] and Islam Md Monirul[4]

[1] Department of Computer Science, University of Science and Technology Bannu, Bannu 28100, Pakistan
[2] Department of Computer Science, Qurtuba University of Science and Information Technology, Peshawar 25000, Pakistan
[3] Interdisciplinary Research Centers for Finance and Digital Economy, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia
[4] College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China

Abstract

Part-of-speech (POS) tagging—the automatic assignment of grammatical categories to every token in a text corpus—is a foundational preprocessing step for AI-driven language applications such as machine translation, sentiment analysis, and information retrieval. For morphologically complex, low-resource languages such as Pashto, the scarcity of annotated data and standardised tools makes this task particularly challenging. This paper presents a systematic comparative evaluation of six machine learning (ML) and deep learning (DL) algorithms—Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP), and Naïve Bayes (NB)—on a newly constructed 32,000-token CoNLL-formatted Pashto corpus (PashtoPoSTags). Each model is assessed by token-level accuracy, the standard metric for POS tagging evaluation. Decision Tree achieved the highest accuracy of 94.34%, followed by KNN with Jaccard distance (94.14%) and KNN with Euclidean distance (93.97%). Random Forest and SVM both exceeded the 90% threshold, while MLP with Tanh activation reached 87.25% and the best Naïve Bayes variant (Complement NB) attained 83.96%. Results show that classical ML algorithms with carefully engineered n-gram features provide computationally efficient and interpretable baselines for POS tagging in resource-constrained settings, and that PashtoPoSTags offers a reproducible benchmark for future Pashto NLP research.

Keywords: machine learning, part of speech tagging, morphological structure, grammatical features.

1 Introduction

In 1950, Allan Turing in a seminar proposed the idea that computers can understand Natural Languages and can communicate with Humans. From the beginning of the twenty-first century, this vision has

Submitted: 13 March 2025
Revised: 02 May 2025
Accepted: 08 May 2025
Published: 30 June 2025

Vol. 1, **No.** 2, 2025.
10.62762/TACS.2025.493945

***Corresponding authors:**
✉ Fida Muhammad Khan
fida5073@gmail.com
✉ Hazrat Bilal
hbilal@mail.ustc.edu.cn

Citation
Khan, A. A., Khan, W., Khan, M. A., Khan, K., Khan, F. M., Rahman, A. U., Bilal, H., & Monirul, I. M. (2025). Comparison of Machine Learning and Deep Learning Models for Part-of-Speech Tagging. *ICCK Transactions on Advanced Computing and Systems*, 1(2), 106–116.

© 2025 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

begun to take on greater plausibility [1]. Natural language processing (NLP) is a subfield of computer science that involves computer programs that evaluates, synthesizes, and tries to understand one or more human languages. Text, spoken language, and keystroke will be used as input. The basic task of NLP is to translate one language into another, generate summaries, and manage dialogue with users as a part of information retrieval [2]. Some problems that require the processing of Natural Language have different relevance. Modern NLP systems draw on a range of techniques—from classical machine learning classifiers to deep neural architectures—to address tasks such as POS tagging, sentiment analysis, and machine translation, and recent work has extended these methods to low-resource languages including Pashto [3]. In the present day and the future, NLP systems will undoubtedly play an important role in human-machine and machine-machine communication. NLP's role in advancing telecommunication and computer science cannot be underestimated. Comparative studies on machine and deep learning approaches for POS tagging have demonstrated their effectiveness across morphologically rich languages, including Urdu [4]. We use NLP in daily life because our computers and smartphones use many software applications that use NLP applications. For morphologically complex and low-resource languages such as Pashto, recent work has demonstrated that combining deep learning models with sequence labeling frameworks can substantially improve POS tagging performance [5]. POS tagging is a foundational NLP task that underpins applications ranging from information extraction to text summarization, and has been approached through a wide spectrum of machine learning and deep learning methods [6]. Natural Language Understanding, which is a subpart of NLP, is divided into subparts, as their morphology, which includes Syntax, semantics, and pragmatics, where morphology defines the creation of words. Syntax means how the words are grouped to form a sentence, semantics refers to the expression of a sentence, and pragmatics stands for how a statement is used in different scenarios [7]. Recent work on Pashto text preprocessing has demonstrated that CRF-based models can effectively handle the language's unique whitespace inconsistencies and word boundary ambiguities [8].

Part of Speech tagging is the subarea of NLP where the text is divided into words and sentences, and the appropriate Part of Speech tag is assigned to each word [9]. Part of Speech tagging is considered a prerequisite to solving some linguistic problems, which may include Speech-to-text conversion, auto-completion, auto-correction, etc. Tagging words in a given sentence manually is a time-consuming process, but with advancements in NLP, now POS tagging is done automatically using tools such as the NLPashto toolkit, which integrates BiLSTM-based sequence labeling for Pashto [10]. Many words in a language like English are often ambiguous in Part-of-Speech tagging. Hence, a Part-of-Speech tagger is a system that assigns a part-of-speech category to words based on contextual information [11]. Assigning a correct Part of Speech tag to a selected word or sentence of any language and then generating new output based on the tagging data, substantial research has been devoted to POS tagging for widely studied languages such as English, Telugu, and Bengali, yet many regional languages with smaller research communities—including South Asian languages such as Urdu and Pashto—remain comparatively underserved, as reflected in the limited progress documented for other South Asian low-resource languages such as Maithili [12]. Part of Speech tagging is essential for Parsing and Grammar correction, etc. Comparative studies of POS tagging techniques for Urdu and other regional languages of Pakistan have demonstrated that dataset quality and annotation consistency are critical determinants of tagger performance [13]. For morphologically rich languages such as Urdu, the POS tagging pipeline involves tokenization, tag assignment using learned models, and disambiguation of words that carry multiple possible tags depending on context [14].

1.1 Pashto Language

Pashto belongs to the Indo-European language family and is primarily spoken in Pakistan and Afghanistan [9]. Recent efforts to build NLP toolkits for Pashto, such as the NLPashto toolkit incorporating BiLSTM-based models, have demonstrated that deep learning approaches can achieve competitive POS tagging accuracy for this low-resource language [10]. Pashto is a low-resource language, and the construction of large, organized datasets demands considerable effort from researchers, as illustrated by corpus-building efforts for tasks such as Pashto sentiment analysis [15]. Researchers take a lot of time in collecting and organizing the dataset of the Pashto language. The Pashto language script is similar to Arabic and Persian and is written from right to left. There is no concept of lower and upper case in the

Pashto language, it is not necessary to add a blank space between two just like in English. Sometimes the space insertion may generate errors or an ambiguous word [16].

As we mentioned earlier, Pashto is a Low-resource language; therefore, it did not gain more attention from the researchers, and hence, very little work was done on the Pashto Language. Previously, some researchers worked on the Pashto language by using various statistical and rule-based methods for POS tagging. A rule-based POS tagging is used by [7] which achieves an accuracy of 88%. A Bi-LSTM with CRF used by [9] and achieve 87.60% of accuracy. In this research study, we focus on using machine learning and deep learning models for POS tagging of the Pashto language. These algorithms include Support Vector Machine, Decision Tree, Random Forest, K-NN, Multi-Layer Perceptron, and Naïve Bayes. The main objective of this research work can be applied to machine translation, sentiment analysis, and Pashto speech recognition. Additionally, Pashto is a low-resource language that is maintained and advanced in the field of computational linguistics.

This paper assesses various Machine Learning and Deep Learning models for POS tagging and offers a well-annotated dataset to alleviate the dearth of resources in Pashto. By enabling applications like sentiment analysis, machine translation, and text-to-speech systems, these contributions can aid in the development of NLP tools for Pashto, which are crucial for the preservation and advancement of the language in digital fields.

The Pashto language has unique grammatical features that make POS tagging challenging, including complex morphology, flexible word order, and the use of diacritical symbols. Additionally, other Pashto dialects add variations in syntax and vocabulary, including Quetta, Yousafzai, Kandahari, and others. These characteristics highlight the importance of this study and make Pashto POS categorization more challenging.

1.2 Challenges

Whenever someone develops a POS tagger for any language, it may face some challenges. The following are some challenges related to the Pashto language.

1.2.1 Ambiguity

Some words in the Pashto language can have different meanings according to sentence structure and context. As shown in Table 1, the Pashto word "کله "means "when," and the other meaning is "Village."

Table 1. Ambiguity example.

کوم	خوب/Sleeping	زه
	I am sleeping	
VB	NN	I-PR

1.2.2 Insufficient resources

Pashto is a low-resource language with limited digital resources; even basic script processing tools such as optical character recognition systems remain under-studied, as demonstrated by Uddin et al. [17], who constructed a benchmark Pashto handwritten character dataset and developed a deep neural network-based OCR system for the Pashto script. The scarcity of annotated Pashto corpora extends across NLP sub-tasks: Iqbal et al. [15] had to construct a dedicated Pashto social media corpus from scratch when developing sentiment analysis models, underscoring that data collection remains a fundamental bottleneck for Pashto NLP research. The lack of annotated datasets and the scarce application of deep learning and sophisticated machine learning techniques for Pashto point-of-sale tagging serve as research gaps that are addressed in this paper. This gap is notable because it hinders the development of NLP tools for Pashto, a low-resource language with rising computational demands.

1.2.3 Lack of Standardization

As shown in Table 2, Pashto has many dialects, this dialect leads to morphological, syntactical, and spelling variation.

Table 2. Dialect variation example.

اولدو	خوب/Dream	ما
	I saw a dream	
VB	NN	I-PR

1.2.4 Research Questions

The main research issue of this work is: How can machine learning and deep learning techniques be applied to accurately perform part-of-speech tagging for the Pashto language, considering its unique grammatical features and resource limitations? This question was used to assess the effectiveness of six approaches in addressing the challenges associated with Pashto POS tagging. Additional questions include:

1. How accurate and effective are machine learning versus deep learning models for Pashto

part-of-speech tagging?

2. Which preprocessing techniques are most effective at handling the unique language traits and script variations of Pashto in POS tagging tasks?
3. Which machine learning or deep learning model is most suited to address the challenges caused by the lack of annotated Pashto datasets for POS tagging?

2 Related Work

Part of speech tagging started in the era of 1950s and 1960s and is performed for many languages, which are known as rich resource languages like English, French, Arabic Chinese, etc. In this article, we focus on PoS tagging for the Pashto language.

The earliest rule-based approach is used by [7], followed the EAGLE guidelines developed a tag set consisting of 54 different tags, and achieved an accuracy of 88%. Another study was conducted by [16], where the authors introduce the sentence structure of the Pashto language, and describe that the sentence is in the form of Subject-Object-Verb or Object-Subject-Verb. In Pashto, many words have more than one meaning, and need attention, a study conducted by [9] do focus on that problem using CRF-LSTM and HMM models, where they achieve accuracies of 87 and 78% respectively. The neural network model BiLSTM was used by [10] with GloVE embedded model, which shows an accuracy of 97%.

Urdu is a widely spoken and national language of Pakistan. Khan et al. [18] developed a CRF-based Urdu POS tagger with language-independent features, achieving state-of-the-art performance and outperforming SVM-based baselines For Arabic, AlKhwiter and Al-Twairesh [19] applied CRF and bidirectional LSTM models to POS tagging of tweets covering both Modern Standard Arabic and the Gulf dialect, achieving up to 96.5% accuracy with the Bi-LSTM approach. Alrajhi and ELAffendi [20] applied deep learning LSTM models to Arabic POS tagging using the Quranic Arabic Corpus, demonstrating that neural approaches can substantially outperform word embedding baselines such as Word2Vec for morphologically rich Semitic languages. Arabic morphology is rich, and its NLP processing requires joint treatment of tokenization, part-of-speech tagging, and morphological disambiguation. Habash and Rambow [21] demonstrated that these three tasks can be addressed simultaneously in a single pipeline, achieving state-of-the-art performance on Modern Standard Arabic by exploiting the interdependencies among clitic segmentation, POS assignment, and morphological features such as gender and voice. The study done by [22] for the Persian language used the Hidden Markov Model on homogenous and heterogenous Persian corpora and achieved 98.1% accurate results. A statistical part-of-speech tagger for the Persian language was developed by Seraji [23], demonstrating the applicability of corpus-based probabilistic approaches to morphologically rich languages closely related to Pashto in script and structure. The study effort [9] uses the Long-Short-Memory (LSTM) Neural Network technology of the deep learning approach to reliably tag words that have several meanings in the sentence; hence, the same word has multiple POS tags. The CRF-BLSTM and HMM models are used by the authors to train the LSTM Neural Network and confirm its accuracy. They get 78% accuracy for HMM and 87% accuracy for CRF-BLSTM.

The literature study reveals the importance of POS tagging for Pashto Language in NLP and will be used in many applications like sentiment analysis and machine translation, etc., as Pashto has a rich morphology, various dialects, and less availability of well-annotated datasets, making the POS task challenging. This research will help to address the issues. POS tagging for low-resource languages presents shared challenges across architectures; Baishya and Baruah [24] specifically investigated the choice of activation functions in deep learning networks for POS tagging under minimal training data conditions, a finding directly relevant to the MLP experiments conducted in the present study.

Part of speech tagging is important for machine translation and many more applications related to NLP; this research study focuses on the development of a POS tagger for the Pashto language.

3 Methodology

The proposed methodology is composed of several machine learning and deep learning models to achieve the task of Part of Speech tagging for the Pashto language, the machine learning models we use in our experiments are Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN) classifier, Naïve Bayes and the deep learning model Multi-layer Perceptron (MLP). For more convenient and efficient results, two distance

methods of the KNN classifier are used, which are the Jaccard and Euclidean distance methods. in Multi-Layer Perceptron, four activation functions, namely Tanh, Identity, ReLu, and Sigmoid, are included in these experiments, and four different variants of Naïve Bayes algorithm, ComplementNB, MultinomialNB, BernoulliNB, and GaussianNB, are also tested in the said experiments.

The following methods were picked because of their shown efficacy in text categorization tasks. While MLP with different activation functions was utilized to investigate nonlinear correlations in the data, SVM, RF, and DT are recognized for managing challenging decision limits. Because of their ease of use and effectiveness when working with small, low-resource datasets like Pashto, KNN and Naive Bayes (with its four variations) were included.

The unique contribution of this research study is to develop a well-annotated dataset, applying six different machine learning and deep learning models with their different variants and activation methods, and a comparison of these models.

3.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is used to find the optimal hyperplane that separates the data items in classes. SVM is suitable for classification and regression problems. Given training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $\mathbf{x}_i$ represents the feature vector and y_i is the class label, the SVM solves the following optimisation problem:

$$\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

subject to

$$y_i\Big(\mathbf{w}^\top\mathbf{x}_i + b\Big) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n \tag{2}$$

where C is the regularisation parameter (set to $C = 1.0$) and ξ_i are slack variables that permit margin violations. We use the one-vs-rest multiclass strategy with an RBF kernel.

3.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a supervised machine learning algorithm that can be used for classification and Regression problems. KNN makes predictions based on similarity between data points. For classification problems, KNN assigns the most relevant label among the neighbors. For regression, it assigns the target by averaging the values. Different distance methods are used to calculate the distance between the neighbors, including:

3.2.1 Euclidean Distance

The Euclidean Distance method is a well-known distance method in geometry. This method is used to find the straight-line distance between the two data points in the Cartesian system. This method is most commonly used in KNN. If we have two data points (X_1, Y_1 and X_2, Y_2) in a 2D space, then the Euclidean distance is calculated as:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

3.2.2 Jaccard Distance method

The Jaccard Distance method in mathematics is used to find the dissimilarity between/Among the sets, and the method is based on the Jaccard Coefficient. This method uses distance ranges that are 0 and 1. If the distance range is 0, it indicates that the sets are identical; if the range is 1, it indicates that there is nothing common between the sets. The Jaccard method is also used in KNN for similarity measurement, it is used with categorical or binary data. For two data points, the Jaccard method is calculated as:

$$\text{Jaccard Distance} = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

This method is also extended to three, four, and other high dimensions. If we have three data points i.e, A, B, and C, then the Jaccard distance will be calculated as:

$$\text{Jaccard distance for AB} = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

$$\text{Jaccard distance for AC} = 1 - \frac{|A \cap C|}{|A \cup C|} \tag{6}$$

3.3 Decision Tree

A Decision Tree is a supervised Machine learning approach used for classification and regression problems, but most commonly, a Decision Tree is used for classification problems. It's called a decision tree because it takes conditions, and based on the condition, it grows like a tree, where Decision nodes and Leaf nodes are generated. The decision nodes take some condition and generate leaf nodes, leaf nodes represent the output of the tree. We use the Gini impurity criterion:

$$G = 1 - \sum_{c=1}^{C} p_c^2. \tag{7}$$

where p_c is the proportion of class c at a given node. Maximum tree depth is tuned on the validation set.

3.4 Random Forest

Random Forest is another popular Machine learning algorithm; it's based on the Decision Tree with the introduction of randomness to improve the performance and decrease overfitting problems. Random Forest is a versatile algorithm and has many application areas, including health care, image identification and recognition, finance, etc. It combines the output of many decision trees and generates a single output, hence called a Random Forest. We use T=100 trees with the Entropy (information gain) split criterion; feature subsampling uses $\sqrt{d}$ features per split, where d is the total feature dimensionality.

3.5 Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a neural network that comes into the category of deep learning algorithms. MLP is considered the base or building block for many other artificial neural networks and can be used for classification, pattern recognition, and regression problems. The structure of an MLP consists of nodes that may be arranged in multiple layers. These layers are divided into three categories, i.e., the Input layer, hidden layers, and Output layer. The data is fed to the input layer of the Network with some weights and bias values, which may be passed to the hidden layer (which may be one or more in number), and finally, the output is displayed on the output layer. The output of layer l is:

$$\mathbf{h}^{(l)} = \sigma\Big(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\Big) \tag{8}$$

where $\sigma(\cdot)$ denotes the activation function. We adopt an architecture with two hidden layers (100 and 50 neurons) and evaluate four activation functions. Two methods, forward and Backward propagation, are used to train the model. In forward propagation, the data is fed to the network through the input layer, and the output is generated layer by layer. In backward propagation, the error is calculated backward, and to remove the error, the Weight and bias values are modified. Different activation functions are used in MLP for processing the input.

3.5.1 *ReLU Activation Function*

ReLU is among the most often used activation functions. It has the following formula:

$$f(m) = \max(0, m) \tag{9}$$

Stated differently, it outputs the input value if it is positive and zero otherwise. The "dying ReLU" problem, in which neurons can become stuck during training and cease updating their weights if they persistently produce negative values, can occur even if the ReLU function adds non-linearity.

3.5.2 *Tanh Activation Function*

Another popular activation function that shifts input values into the $[-1, 1]$ range is Tanh (Hyperbolic). The formula:

$$f(Y) = \frac{e^{Y} - e^{-Y}}{e^{Y} + e^{-Y}} \tag{10}$$

Tanh has the advantage of being zero-centered and introduces non-linearity, both of which can help with model optimization and mitigate some of the problems with ReLU.

Table 3. Symbols excluded from the dataset.

Punctuation Marks	Special Characters	Other Symbols
Question Mark (?)	Semicolon (;)	Ellipsis (…)
Exclamation Mark (!)	Hyphen (-)	Ampersand (&)
Comma (,)	Apostrophe (')	Dollar Sign ($)
Period/Full Stop (.)	Quotation Marks (" ")	Percent Sign (%)
Colon (:)	Parentheses ( )	Plus Sign (+)
Minus Sign (-)	Equal Sign (=)	Greater Than (>) and
Less Than (<) Signs		
At Sign (@)	Asterisk (*)	And others

3.5.3 *Sigmoid (Logistic) Function*

The sigmoid activation function assigns input values to a range (0, 1). It may be calculated using the formula

$$f(c) = \frac{1}{1 + e^{-c}} \tag{11}$$

Despite being widely used in the past, the sigmoid function has many disadvantages, such as disappearing gradients, which can make training deep networks difficult.

3.6 Naïve Bayes

Naïve Bayes is a popular machine learning algorithm, based on the Bayes Theorem of probability. It is called "naïve" because it considers that all the features

should be independent. This algorithm is widely used for classification problems due to its efficiency and simplicity. It's also known as a Probabilistic classifier. Other variants of Naïve Bayes classifier are Multinomial Naïve Bayes, Gaussian Naïve Bayes, Bernoulli Naïve Bayes, and Complement Naïve Bayes. The Naïve Bayes algorithm is based on Bayes' theorem of probability, mathematically expressed as

$$P(H|E) = \frac{P(E|H)^* P(H)}{P(E)} \tag{12}$$

3.7 Proposed Dataset

A carefully annotated dataset with words matched to their POS tags is needed by academics to perform part-of-speech tagging efficiently. Because there isn't a standardized dataset for the Pashto language, we created PashtoPoSTags, our dataset that we annotated with the help of Pashto language specialists by combining data from several sources. To make it easier to apply Machine Learning algorithms correctly, we then converted the dataset into CoNLL format. Sentences are indicated by blank lines to indicate gaps between sentences in the CoNLL format, which divides the data into two columns: one for Pashto words and the other for their matching POS tags. The data was put through several preprocessing steps to ensure its quality and fit for the models. Tokenization was used to separate the text into discrete words, normalization was employed to manage variations in Pashto script, and missing values were dealt with by removing incomplete entries. The dataset was ensured to be clean and ready for analysis by taking these steps.

3.7.1 Data Pre-processing

Before using data for the experiment, the data must be pre-processed and prepared for further processing. The following pre-processing steps are involved. The dataset is presented in both TXT and CSV formats, as shown in Figure 1.

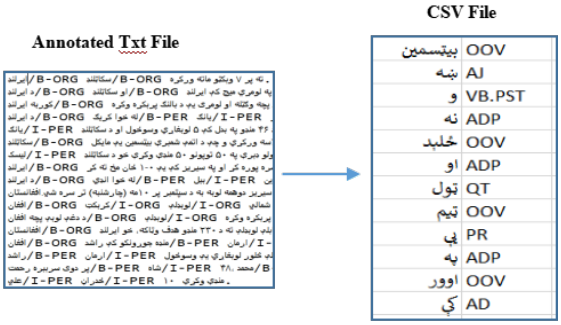


Figure 1. The dataset in TXT and CSV file formats.

Character Encoding The dataset we built consists of words and their corresponding POS tags, which are alphabets, numbers, and symbols, converted into Binary (stream of 0's and 1's) to understand by a computer. For this, we use the "UTF-8" character encoding technique.Table 3 illustrates the symbols excluded from the dataset.

Data cleaning Removing unnecessary symbols and letters is obligatory because it can cause ineffective results, as we arrange the dataset in CoNLL format, hence blank spaces are there in the dataset as well, which are known as NaN values in Python. So, the following symbols that are not part of the language are removed during this step. As shown in Table 4, the unnecessary symbols and characters are removed during the data cleaning process to ensure the dataset is properly formatted for further analysis.

Table 4. Example of data cleaning.

Original Sentence	Cleaned Sentence	Tokenized Sentence	POS Tags
زه کور ته ځم	زه کور ته ځم	زه، کور، ته، ځم	[PR, NN, ADP, VB]

Dataset Splitting The dataset was divided into training, validation, and testing sets in a standard 70/15/15 ratio to ensure accurate model evaluation. Accuracy was the primary evaluation factor used to compare model performances consistently across all experiments. These steps ensured that the models were trained and evaluated systematically.

Lexicon/Lexemes/tokenization. After data cleaning, the dataset is split into tokens. in training, each token consists of a single word with a corresponding POS tag. When testing, the word is checked by the machine learning algorithms and assigned an appropriate POS tag.

"PashtoPoSTags" is a CoNLL-formatted dataset with two columns: one for words and another for the associated POS tags. To guarantee linguistic diversity, the dataset includes 32000 samples from various sources, such as news articles, books, and social media posts. For reproducibility and ease of integration with POS tagging techniques, this structure provides a standard format.

4 Experiments

4.1 Evaluation Metrics

Model performance is quantified by token-level accuracy on the held-out test set, defined as the

proportion of tokens assigned the correct POS tag:

$$\text{Accuracy} = \frac{\text{Number of correctly tagged tokens}}{\text{Total number of tokens}} \tag{13}$$

Accuracy is the standard primary metric for POS tagging evaluation and enables direct comparison with prior Pashto POS tagging studies [7, 9, 10].

The models mentioned in section 3 are applied by conducting the following pattern for experiments. The dataset we prepared for the experiment, named "PashtoPoSTags," is used in our experiments. To feed data into a deep learning or machine learning model, the dataset must be split into separate training and testing subsets. Within this framework, the PashtoPoSTags dataset is partitioned so that 70% of the data is set aside for training, 15% is set aside for testing purposes, and the last 15% is for validation. All experiments were implemented in Python 3.10 using `scikit-learn` 1.3 for the ML models and `TensorFlow` 2.13 for the MLP. Feature extraction used `scipy` sparse matrices; experiments were run on an Intel Core i7-12700H CPU with 16 GB RAM. No GPU acceleration was required for the classical ML models; MLP training completed in under 30 minutes.

Among all the methods we examined in our trials, the Decision Tree machine learning model produced the best accuracy of 94.34% when we used the dataset we assembled as input. This result highlights how well Decision Trees can identify trends in data, which makes it easier to classify POS tags for Pashto words in an effective manner.

In this experiment, we look at two distance approaches that provide two distinct accuracies, and the KNN algorithm demonstrates robustness. While the Euclidean Distance approach yields an accuracy of 93.97%, KNN with the Jaccard Distance method generates an accuracy of 94.14%. The selection of nearest neighbors has a big impact on the KNN algorithm's output; in certain situations, selecting fewer nearest neighbors yields good results, while in other situations, selecting more nearest neighbors may yield good results. We use a random selection process to see how the KNN reacts in our experiments. we noticed that when the number of nearest neighbors is changed, the KNN yields varied results. The neighbourhood size is set to k=3 after grid search over $k \in \{1, 3, 5, 7, 9\}$ on the validation set. Figure 2 illustrates the comparison between Decision Tree and K-Nearest Neighbor (KNN) models.

The method used to estimate the distance between

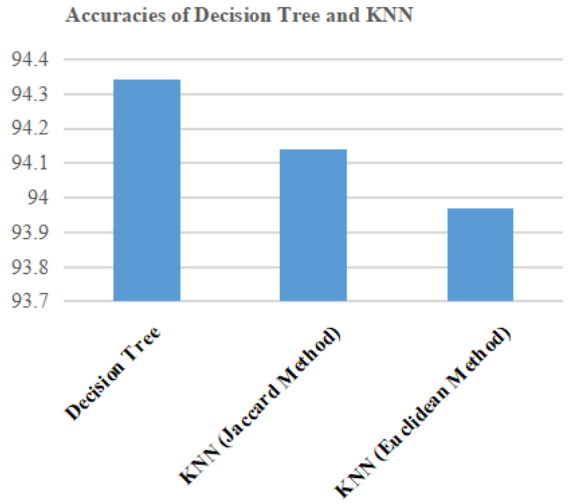


Figure 2. Comparison of Decision Tree & KNN.

the data points is another important factor that could influence the result. This distance can be measured using many techniques, such as the Jaccard, Manhattan, Cosine, and Euclidean distance approaches, as we indicated in the Introduction section. Every approach for measuring distance has a unique outcome. In these investigations, we used two distance methods: the Euclidean and the Jaccard, and each yielded result that were different from the other.

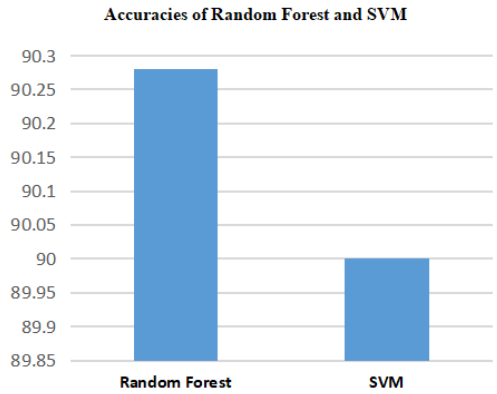


Figure 3. Comparison of Random Forest & SVM.

While the SVM produces results with an accuracy of 90.00%, the Random Forest tests yield results with an accuracy of 90.28%. Even while Decision Tree and KNN (with both distance methods) have slightly higher accuracy than RF and SVM, these algorithms nevertheless provide competitive performance for Pashto language POS tagging. While Random Forest blends several decision trees to get the output, making it a useful method for some applications, SVM is good at distinguishing data points in high-dimensional space. Figure 3 shows the comparison of Random Forest and Support Vector Machine (SVM) in terms of accuracy.

Table 5. Details of feature analysis for different machine learning models.

Model	Feature Strategy	Configuration
KNN	N-gram TF-IDF (2–4), word bigrams	$k = 3$; Euclidean and Jaccard distance
Decision Tree	N-gram TF-IDF (2–4), word bigrams	Gini impurity; depth tuned on validation set
SVM	N-gram TF-IDF (2–4), word bigrams	RBF kernel; $C = 1.0$; one-vs-rest
Random Forest	N-gram TF-IDF (2–4), word bigrams	Entropy; $T = 100$ trees
MLP	N-gram TF-IDF (2–4), word bigrams	Layers [100, 50]; ReLU/Tanh/Sigmoid/Identity; batch=32
Naïve Bayes	N-gram TF-IDF (2–4), word bigrams	Multinomial, Gaussian, Bernoulli, Complement

The accuracy of the Multilayer Perceptron (MLP) varies based on the activation functions that are used. Popular activation functions include Tanh, identity, sigmoid, and ReLU, and various accuracies are obtained. More specifically, the accuracy is 86.65% when using the sigmoid and ReLU activation functions. With the Identity activation function, the accuracy is 86.99%, which is marginally higher. Out of all the activation functions studied, the Tanh function yields the highest accuracy of 87.25%.

The importance of choosing the right activation function for maximizing the Multilayer Perceptron's (MLP) performance is highlighted by these results. Tanh is the most appropriate option among the tested activation functions for improving accuracy in this POS tagging activity. This proves that activation functions are essential for optimizing the performance of the MLP for certain uses, like part-of-speech tagging. Determining the Multilayer Perceptron's (MLP) hidden layer configuration, that is, how many layers and how many neurons are in each layer, is a crucial step in optimizing the MLP. We chose a design with two hidden layers for our investigation, the first with 100 neurons and the second with 50 neurons. It's important to note that changing these settings has a substantial impact on both the algorithm's overall accuracy and the effectiveness of each activation function used. Through a series of studies examining various hidden layer configurations and neuron counts, we concluded that the best performance may be achieved by using two hidden layers, each with 100 and 50 neurons.

The MLP model was trained with a batch size of 32, which is commonly chosen to balance computational efficiency and model generalization. The Pashto dataset's modest size was effectively handled by this batch size, which also ensured steady gradient updates throughout training. This specific feature shows how careful consideration was given to enhancing model performance. Figure 4 compares the performance of different activation functions used in the Multi-Layer Perceptron (MLP) model.

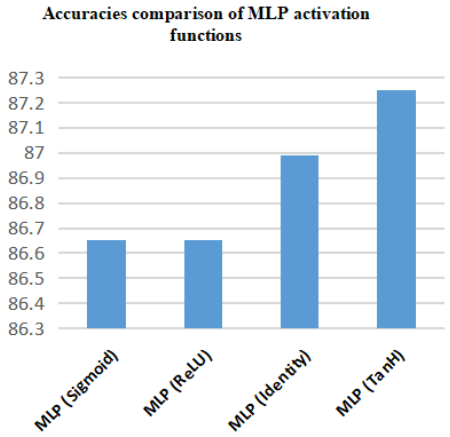


Figure 4. Comparison of MLP activation functions.

There exist numerous iterations of the Naive Bayes algorithm, and their respective accuracy rates vary based on the given circumstances. we employed several different variations in our experiment, such as Gaussian Naive Bayes (GaussianNB), Bernoulli Naive Bayes (BernoulliNB), Complement Naive Bayes (ComplementNB), and Multinomial Naive Bayes (MultinomialNB). In contrast, the accuracy produced by Multinomial Naive Bayes is 80.33%; Gaussian Naive Bayes is 63.98%; Complement Naive Bayes is 83.96%; and Bernoulli Naive Bayes is 73.45% of the course. These results indicate that using a different form of the Naïve Bayes algorithm for the same example can yield different results. Figure 5 compares the performance of Naive Bayes and its different variants.

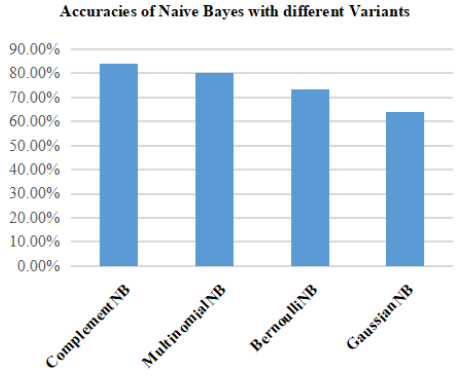


Figure 5. Comparison of Naive Bayes and its different variants.

The underlying assumptions of each version and the characteristics of the dataset are what cause the variations in the accuracy of Naive Bayes variants. The assumption of continuous data and a normal distribution in Gaussian Naive Bayes may not be well adapted to the dataset's discrete word-based characteristics. On the other hand, because Complement Naive Bayes is designed for imbalanced datasets, it is more ideal for controlling the word distributions in Pashto POS tagging.

The Naive Bayes algorithm's results demonstrate how versatile it is, as different variants produce different results on the same dataset. Despite this flexibility, it performs the worst out of the six tested algorithms. This demonstrates that the algorithm can handle a wide range of data distributions, but it also shows that it is not as effective as other approaches in this specific experiment. As shown in Table 5, the different machine learning models used in this study vary in their feature analysis methods and configuration settings, such as the KNN's use of Euclidean distance and Decision Tree's reliance on Gini Index.

5 Discussion

5.1 Implications for Advanced Computing Systems

Decision Tree and KNN are interpretable, lightweight, and deployable on edge devices without GPU acceleration-properties of direct value for IoT-embedded Pashto NLP applications (e.g., voice interfaces, SMS filtering, or intelligent document processing). The computational overhead is orders of magnitude lower than transformer-based approaches, making these methods viable for systems with strict latency or power budgets.

5.2 Limitations

Three limitations should be acknowledged. First, the 32,000-token corpus is modest; the performance gap to neural methods is likely to close as data scales. Second, model accuracy was computed on a single stratified train/test split; future work should adopt k-fold cross-validation to obtain variance estimates. Third, the feature set does not incorporate contextual word embeddings.

5.3 Future Directions

Future work should: (i) expand PashtoPoSTags to $\geq$100,000 tokens; (ii) incorporate mBERT/XLM-R representations; (iii) evaluate cross-dialect generalisation; and (iv) integrate the tagger into a full NLP computing pipeline with downstream parsing and sentiment analysis.

6 Conclusion

In this research study, we examine different machine learning and deep learning algorithms for Pashto language Part of Speech Tagging (POST), which are Decision Tree, Support Vector Machine, Random Forest, Multi-Layer Perceptron with different activation functions like Tanh, Sigmoid, ReLu and Identity, K-Nearest Neighbor by examining two distance methods Jaccard and Euclidean and Naïve Bayes with its four different variants. Among all these algorithms, we assess Decision Tree and KNN with the Jaccard and Euclidean methods, generating the highest accuracies of 94.34%, 94.14%, and 93.97%, respectively. However, this research needs more focus on some parameters, such as the size and diversity of the dataset, and implementing deep learning and ensemble methods, which may produce different results in this case. KNN and Decision Trees performed better because they can manage non-linear decision boundaries and successfully adjust to the structure of the Pashto dataset. Naïve Bayes performed badly because its strong independence assumptions are not as well adapted to the complex relationships present in language data. Notwithstanding its power, MLP's accuracy was reduced due to the tiny quantity of the dataset, which may have limited its ability to fully comprehend the patterns in Pashto POS tagging. Future research could visualise the learned decision tree structure to improve interpretability, offering insights into how the model prioritises particular linguistic features.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current

trends and challenges. *Multimedia tools and applications, 82*(3), 3713-3744. [Crossref]

[2] Lauriola, I., Lavelli, A., & Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing, 470*, 443-456. [Crossref]

[3] Ullah, S., Ahmad, R., Namoun, A., Muhammad, S., Ullah, K., Hussain, I., & Ibrahim, I. A. (2024). A deep learning-based approach for part of speech (pos) tagging in the pashto language. *IEEE Access, 12*, 86355-86364. [Crossref]

[4] Khan, W., Daud, A., Khan, K., Nasir, J. A., Basheri, M., Aljohani, N., & Alotaibi, F. S. (2019). Part of speech tagging in urdu: Comparison of machine and deep learning approaches. *IEEE Access, 7*, 38918-38936. [Crossref]

[5] Zaman, F., Maqbool, O., & Kanwal, J. (2024). Leveraging bidirectional lstm with crfs for pashto tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing, 23*(4), 1-17. [Crossref]

[6] Chiche, A., & Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data, 9*(1), 10. [Crossref]

[7] Rabbi, I., Khan, M. A., & Ali, R. (2008, March). Developing a tagset for Pashto part of speech tagging. In *2008 Second International Conference on Electrical Engineering* (pp. 1-6). IEEE. [Crossref]

[8] Haq, I., Qiu, W., Guo, J., & Tang, P. (2023). Correction of whitespace and word segmentation in noisy Pashto text using CRF. *Speech Communication, 153*, 102970. [Crossref]

[9] Haq, I., Qiu, W., Guo, J., & Peng, T. (2023). *The Pashto corpus and machine learning model for automatic POS tagging*. Research Square. [Crossref]

[10] Haq, I., Qiu, W., Guo, J., & Tang, P. (2023). NLPashto: NLP toolkit for low-resource Pashto language. *International Journal of Advanced Computer Science and Applications, 14*(6). [Crossref]

[11] Khan, H. A., Ali, M. J., & Hanni, U. E. (2020, November). Poster: A novel approach for pos tagging of pashto language. In *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)* (pp. 259-260). IEEE. [Crossref]

[12] Priyadarshi, A., & Saha, S. K. (2023). A study on the performance of recurrent neural network based models in Maithili part of speech tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing, 22*(2), 1-16. [Crossref]

[13] Rajper, R. A., Rajper, S., Maitlo, A., & Nabi, G. (2021). Analysis and comparative study of POS tagging techniques for national (Urdu) language and other regional languages of pakistan. *SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES), 53*(04). [Crossref]

[14] Naz, F., Anwar, W., Bajwa, U. I., & Munir, E. U. (2012). Urdu part of speech tagging using transformation based error driven learning. *World Applied Sciences Journal, 16*(3), 437-448. https://www.researchgate.net/publication/267964478

[15] Iqbal, S., Khan, F., Khan, H. U., Iqbal, T., & Shah, J. H. (2022). Sentiment analysis of social media content in pashto language using deep learning algorithms. *Journal of Internet Technology, 23*(7), 1669-1677. https://jit.ndhu.edu.tw/article/view/2833

[16] Rabbi, I., Khan, A. M., & Ali, R. (2009). Rule-based part of speech tagging for Pashto language. In *Conference on Language and Technology, Lahore, Pakistan*. https://www.cle.org.pk/clt09/download/Papers/Paper12.pdf

[17] Uddin, I., Ramli, D. A., Khan, A., Bangash, J. I., Fayyaz, N., Khan, A., & Kundi, M. (2021). Benchmark Pashto handwritten character dataset and Pashto object character recognition (OCR) using deep neural network with rule activation function. *Complexity, 2021*(1), 6669672. [Crossref]

[18] Khan, W., Daud, A., Nasir, J. A., Amjad, T., Arafat, S., Aljohani, N., & Alotaibi, F. S. (2019). Urdu part of speech tagging using conditional random fields. *Language Resources and Evaluation, 53*, 331-362. [Crossref]

[19] AlKhwiter, W., & Al-Twairesh, N. (2021). Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM. *Computer Speech & Language, 65*, 101138. [Crossref]

[20] Alrajhi, K., & ELAffendi, M. A. (2019). Automatic arabic part-of-speech tagging: Deep learning neural lstm versus word2vec. *International Journal of Computing and Digital Systems, 8*(03), 307-315. https://pdfs.semanticscholar.org/ddd1/095d99b25026377a8dc0f1af590d377a5168.pdf

[21] Habash, N., & Rambow, O. (2005, June). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)* (pp. 573-580). [Crossref]

[22] Okhovvat, M., & Bidgoli, B. M. (2011). A hidden Markov model for Persian part-of-speech tagging. *Procedia Computer Science, 3*, 977-981. [Crossref]

[23] Seraji, M. (2011, May). A statistical part-of-speech tagger for Persian. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)* (pp. 340-343). https://aclanthology.org/W11-4654/

[24] Baishya, D., & Baruah, R. (2024). Part-of-speech tagging for low-resource languages: activation function for deep learning network to work with minimal training data. *ACM Transactions on Asian and Low-Resource Language Information Processing, 23*(5), 1-31. [Crossref]