IEEE
TÜRKİYE SECTION

UBMK'25

Bildiriler Kitabı
Proceedings

Editör Eşref ADALI

10. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı

10th International Conference on Computer Science and Engineering

17-18-19 Eylül (September) 2025 İstanbul - Türkiye

IEEE TÜRKİYE SECTION

UBMK'25

Bildiriler Kitabı
Proceedings

Editor Eşref ADALI

10. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı

10th International Conference on Computer Science and Engineering

17-18-19 Eylül (September) 2025 İstanbul - Türkiye

Media type	Part Number	ISBN	Online ISSN
XPLORE COMPLIANT	CFP25L97-ART	979-8-3315-9975-1	2521-1641
CD-ROM	CFP25L97-CDR	979-8-3315-9974-4	

10. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK'2025)

10th International Conference on Computer Science and Engineering

17-18-19 Eylül 2025 -İstanbul-Türkiye
17-18-19 September 2025 - İstanbul-Türkiye

IEEE Catalog Number : CFP25L97-ART

ISBN : 978-8-3315-9975-1

Additional copies mey be ordered from:
Curran Associates, Inc
57 Morehouse Lane Red Hook, NY 12571 USA
Phone: (845) 758 0400
Fax: (845) 758 2633
E-mail: curran@proceeding.com

UBMK’2025’ye Hoşgeldiniz
Welcome to UBMK’2025

Sevgili Katılımcılar:

UBMK uluslararası nitelikli konferans serisi, 1990 yılından beri düzenli olarak yapılmakta olan Bilgisayar Mühendisliği Bölüm Başkanları toplantılarında alınan bir kararla on yıl önce başlamıştır. Konferansın 10.su IEEE-UBMK-2025 bu yıl 17-18-19 Eylül, 2025 günlerinde İstanbul Teknik Üniversitesinin ev sahipliğinde düzenlemiştir.

IEEE-UBMK-2025 konferansına bu yıl Almanya, Amerika Birleşik Devletleri, Azerbaycan, Fransa, Irak, İngiltere, İsveç, İtalya, Kanada, Kazakistan, Kırım, Kırgızistan, Rusya, Özbekistan, Tataristan, Taylant, Ürdün ve Türkiye’den 610 dolayında bildiri gönderilmiş ve bu bildiriler Türk ve yabancı 250 hakem tarafından değerlendirilmiştir.

Her bildiri en az iki hakem tarafından incelenmiş ve uzlaşma olmadığı durumlarda üçüncü bir hakemin değerlendirmesine başvurulmuştur. Bildiri başına düşen ortalama hakemlik 2,3 olmuştur. Bu değerlendirmelerin sonunda 327 bildirinin sözlü olarak sunulması uygun bulunmuştur. Kabul edilen ve sunulan bildiriler içerik ve kalite ölçünlerini sağlaması durumunda IEEE Xplore’da yayımlanacaktır.

Konferans çalışmalarında, Bilgisayar Mühendisliği Bölüm Başkanları Danışma Kurulu olarak görev almışlardır. Bildirilerin değerlendirilmesi Bilim Kurulu üyeleri tarafından yapılmıştır. Konferansın düzenlenmesi ise Yürütme Kurulunun önerileri doğrultusunda, Düzenleme Kurulu tarafından yapılmıştır.

Son olarak, konferansın başarılı bir şekilde yürütülmesi için tüm olanaklarını sunan İstanbul Teknik Üniversitesi Rektörü Sayın Prof. Dr. Hasan Mandal’a teşekkür ediyoruz. Ayrıca Düzenleme Kuruluna, bildirileri titizlikle değerlendiren Bilim Kurulu Üyelerine ve değerli araştırmalarının sonuçlarını bilişim camiası ile paylaşan bildiri sahiplerine teşekkürlerimizi iletiriz.

Prof. Dr. Eşref ADALI
UBMK-2025 Konferans Başkanı ve Bildiri Kitabı Editörü

Dear Participants:

The UBMK international conference series started nine years ago with a decision taken at the Computer Engineering Department Heads (BMBB) meetings, which have been held regularly since 1990. The 10th edition of the conference, UBMK'25, was held this year on October 17-18-19, 2025, hosted by İstanbul Technichal University.

This year, approximately 610 papers were submitted to the IEEE-UBMK-2025 conference from Germany, the United States, Azerbaijan, France, Iraq, the United Kingdom, Sweden, Italy, Canada, Kazakhstan, Crimea, Kyrgyzstan, Russia, Uzbekistan, Tatarstan, Thailand, Jordan, and Turkey, and these papers were evaluated by 250 Turkish and foreign referees.

Each paper was evaluated at least by two referees, and in cases where there was no consensus, a third referee was consulted. At the end of these evaluations, 327 papers were accepted for oral presentation. Accepted and presented papers will be submitted for inclusion into IEEE Xplore subject to meeting IEEE Xplore's scope and quality requirements.

During the conference, Heads of Information Engineering Departments took part in the Advisory Board. The evaluation of the papers was made by the members of the Scientific Committee. The conference was organized by the Organizing Committee in line with the recommendations of the Executive Committee.

Finally, we would like to thank İstanbul Technical University Rector Prof. Dr. Hasan Mandal for his continued support for the success of the conference. In addition, we would like to thank the Organizing Committee, the Scientific Committee Members who carefully evaluated the papers, and the owners of the papers who shared the results of their valuable research with the informatics community.

Prof. Dr. Esref ADALI
UBMK'25 Conference Chair and Proceedings Editor

Düzenleyenler / Organizer

Destekleyenler / Sponsors

Statistical POS Tagging Algorithms (HMM, CRF)

Odinakhon Jamoldinova
Dept. of Social Sciences and Humanities
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
odinakhonjamoldinova1970@gmail.com

Botir Boltayevich Elov
Dept. of Computational Linguistics and Digital Technologies
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
elov@navoiy-uni.uz
ORCID: 0000-0001-5032-6648

Maftunakhon Sharipova
Dept. of Uzbek Philology and Journalism
Bukhara State University
Bukhara, Uzbekistan
sharipovamaftuna26@gmail.com

Shakhzoda Miralimova
Dept. Uzbek Philology
Andijan State Institute of Foreign Languages
Andijan, Uzbekistan
shakhzodamiralimova@gmail.com

Zilola Yuldashevna Xusainova
Dept. of Computational Linguistics and Digital Technologies
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
xusainovazilola@navoiy-uni.uz
ORCID: 0000-0003-4357-7515

Nizomaddin Uktambay o'g'li Khudayberganov
Dept. of Computational Linguistics and Digital Technologies
Tashkent State University of Uzbek Language and Literature named Alisher Navo'i
Tashkent, Uzbekistan
nizomaddin@navoiy-uni.uz
ORCID: 0000-0002-6213-3015

Kholmurod Karimov
Dept. of Psychology
Karshi State University
Karshi, Uzbekistan
karimovxolmurod057@jmail.ru

Abstract—This paper presents a comprehensive study of two statistical approaches to part-of-speech (POS) tagging in Uzbek – the Hidden Markov Model (HMM) and the Conditional Random Field (CRF) – from both mathematical and empirical perspectives. We first formalize each model: transition and emission probabilities for HMM, and feature functions with weight parameters for CRF. Both models were trained on a 205 k-token (77821 sentences) CONLL-U corpus annotated with 15 Uzbek-specific POS tags, employing Laplace-smoothed Viterbi decoding for HMM and an L-BFGS-optimized CRF with Viterbi inference. On the held-out test set, the HMM achieved 82 % tagging accuracy, while the CRF reached 88 %, outperforming HMM by six percentage points thanks to its richer contextual and linguistic features. The results confirm that statistical models remain robust for agglutinative, low-resource languages like Uzbek, yet are sensitive to feature engineering. We conclude with an error analysis, guidelines for model selection, and perspectives on migrating to neural architectures such as BiLSTM-CRF and BERT-based taggers.

Keywords — Uzbek language, POS tagging, Hidden Markov Model, Conditional Random Field, statistical NLP, Viterbi algorithm, low-resource language

I. INTRODUCTION

Automatic part-of-speech tagging (POS tagging) for the Uzbek language refers to the task of assigning the correct grammatical category (noun, verb, adjective, etc.) to each word in a text [1]. This task is one of the fundamental problems in natural language processing and serves as a crucial intermediate step for more advanced linguistic analyses, such as syntactic parsing, semantic analysis, information retrieval, and machine translation. In corpus linguistics, texts accurately tagged with part-of-speech information allow for comprehensive research on the structural features of a language.

For resource-rich languages like English and Russian, numerous POS tagging methods have been developed, achieving accuracy levels of nearly 97%. However, for low-resource agglutinative languages such as Uzbek, POS tagging remains a significant challenge.

Nowadays, deep learning-based approaches have introduced new possibilities to traditional statistical models. By integrating neural networks into conventional models such as HMM and CRF, more efficient POS taggers are being developed. For instance, the BiLSTM-CRF architecture [2] (bidirectional LSTM combined with CRF) has proven to be highly effective for sequence tagging, as demonstrated in several studies [1].

Furthermore, transformer-based models pre-trained on large corpora (such as BERT) have enabled significant improvements even for low-resource languages through architectures like BERT-CRF. One of the earliest works on neural network-based POS tagging for Uzbek is the study by Murat and Ali [4]. In their research, they proposed a model that generates deeper word representations via affix information and incorporates a multi-head attention mechanism. This model achieved a 4.13% improvement in accuracy compared to traditional BiLSTM, CNN, and CRF models, reaching an overall accuracy of 79.74%.

More recently, Bobojonova et al. (2025) developed a dedicated corpus and models for BERT-based POS tagging in Uzbek, named BBPOS, and reported an average accuracy of 91%[5].

This paper presents the mathematical foundations and operational principles of statistical POS tagging models for the Uzbek language[6]. First, we provide a formal definition of the HMM and CRF models, along with a detailed explanation of their deep learning-based extensions (i.e., classical CRF vs. BiLSTM-CRF vs. BERT-CRF). The mathematical structures of each model are represented through equations, and their algorithms are described using pseudocode. Subsequently, the models are trained and evaluated on a POS-tagged corpus of Uzbek sentences in CoNLL-U format, consisting of 6,528, 56,616, and 77,821 sentences, respectively. We then conduct a comparative analysis of the performance of these models.

II. Methods

A. POS Tagging Based on Hidden Markov Model (HMM)

An HMM is a generative statistical model for sequential data, which models the probability that an observed sequence of words has been generated from a sequence of hidden states (i.e., tags)[7]. In the context of POS tagging, the HMM assumes the following stochastic process: at each position in a sentence, a particular tag (e.g., noun, verb, adjective, etc.) is chosen based on the previous tag according to the rules of a Markov chain (the sequence of hidden states), and then a word corresponding to that tag is "emitted" at that position.

Thus, the HMM model involves two types of probability parameters: **transition probabilities** and **emission probabilities**

- **The transition probability** $\boldsymbol{P(t_i|t_{i-1})}$ represents the likelihood of a tag t_i occurring after the previous tag t_{i-1}. For example, the probability of a verb following a noun, or a postposition following a verb.

To estimate these probabilities, the model relies on the **first-order Markov assumption**, meaning that each tag depends only on the immediately preceding tag, not on any earlier ones. In the training corpus, these probabilities are computed from the frequencies of tag pairs as follows:

$$\boldsymbol{P(t_i|t_{i-1}) = \frac{C(t_{i-1},t_i)}{C(t_{i-1})}} \quad (1)$$

here,

$\boldsymbol{C(t_{i-1}, t_i)}$ – the number of times tag $\boldsymbol{t_i}$ occurs consecutively after tag $\boldsymbol{t_{i-1}}$

$\boldsymbol{C(t_{i-1})}$ – the total number of occurrences of tag $\boldsymbol{t_{i-1}}$ in the corpus.

The **emission probability** is expressed as $\boldsymbol{P(w_i|t_i)}$, representing the probability of observing the word $\boldsymbol{w_i}$ given the tag $\boldsymbol{t_i}$. This probability is also estimated based on frequencies in the corpus:

$$P(w_i|t_i) = \frac{C(t_i,w_i)}{C(t_i)} \quad (2)$$

here,

$\boldsymbol{C(t_i, w_i)}$ – the number of times the word $\boldsymbol{w_i}$ occurs with the tag $\boldsymbol{t_i}$.

$\boldsymbol{C(t_i)}$ – the total number of occurrences of the tag $\boldsymbol{t_i}$ in the corpus. These probabilities are stored in matrix form as parameters: **A (transition probabilities)** and **B (emission probabilities)**.

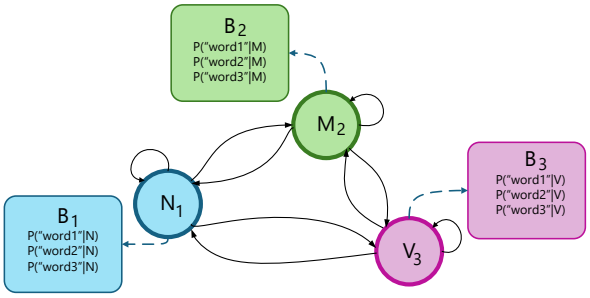

Fig. 1. **Example of a Hidden Markov Model** (with three hidden states: N – noun, M – modal verb, V – verb)

The HMM model assumes that the hidden tag sequence behind the given word sequence *"word1 word2 word3"* is N–M–V. In **Fig 1**, the blocks **B1**, **B2**, and **B3** show the emission probabilities from each state to a specific word. For example, block B1 includes values such as $\boldsymbol{P("word1"|N)}$. The arrows between the states represent the **transition probabilities** between tags, as defined in the transition matrix **A**.

The goal of the HMM model is to find the most probable sequence of tags $\boldsymbol{\hat{T} = t_1, t_2, \dots, t_n}$ for a given sequence of observed words $\boldsymbol{W = w_1, w_2, .., w_n}$ such that the joint probability $\boldsymbol{P(W, T)}$ is maximized.

As a generative model, the joint probability P(W,T)P(W, T)P(W,T) is computed as follows:

$$P(W|T) = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \quad (3)$$

here, $\boldsymbol{t_0 = <Boshlanish>}$ is assumed to be a special start state.

Exhaustively checking all possible tag combinations to find the most probable tag sequence **T** is computationally infeasible, as it would require evaluating $O(m^n)$ combinations (where mmm is the number of possible tags and m is the number of words).

To address this, the **Viterbi algorithm**, based on dynamic programming, is employed. The Viterbi approach recursively computes the most probable path to each tag at every position, and ultimately determines the overall optimal tag sequence.

Below is the pseudocode of the Viterbi algorithm for POS tagging using an HMM model.

Algorithm 1: Viterbi Decoding Based on HMM (for POS Tagging)

Input:
w[1..*n*] – *sequence of words to be tagged*
Output:
t[1..n] – most probable sequence of POS tags
1 *# 1. Initialization – transition from Start state to the first tag:*
2 **for** *each tag* ***s***:

3 $\delta[1][s] = P(s \mid < Start >) * P(w_1 \mid s)$
4 $backpointer[1][s] = < Start >$
5 *# 2. Recursive step – compute the best path to each state:*
6 **for** i *from 2 to n*:
7 **for** *each tag* $\boldsymbol{s}$:
8 *# find the highest probability path from the previous position to state s*
9 $max_{prob} = 0$
10 $arg_{max_{state}} = None$
11 **for** *each tag* $\boldsymbol{s_{prev}}$:
12 $prob = \delta[i-1][s_{prev}] * P(s \mid s_{prev}) * P(w_i \mid s)$
13 **if** $prob > max_{prob}$:
14 $max_{prob} = prob$
15 $arg_{max_{state}} = s_{prev}$
16 $\delta[i][s] = max_{prob}$ *# Best score up to position i*
17 $backpointer[i][s] = arg_{max_{state}}$ *# Best previous tag for position i*
18 *# 3. Termination – find the tag with the highest probability at the last position:*
19 $max_{prob} = 0$
20 $last_{state} = None$
21 **for** *each tag* $\boldsymbol{s}$:
22 **if** $\delta[n][s] > max_{prob}$:
23 $max_{prob} = \delta[n][s]$
24 $last_{state} = s$
25 *# 4. Backtrace – reconstruct the best tag sequence:*
26 $t[n] = last_{state}$
27 **for** i *from n down to 2*:
28 $t[i-1] = backpointer[i][\, t[i]\,]$
29 **return** $t[1..n]$

In the above algorithm, the value $\boldsymbol{\delta[i][s]}$ represents the highest probability of a partial path ending in tag s at position iii, given the word sequence $\boldsymbol{w_1}$ ... $\boldsymbol{w_i}$. The backpointer arrays store pointers to previous tags in order to reconstruct the optimal tag sequence through backtracking.

The time complexity of the algorithm is $\boldsymbol{O(n \times m^2)}$, where m is the number of tags and n is the number of words.

As a result, CRF models rely on fewer independence assumptions and can incorporate a wide range of correlated features

For this reason, CRFs evaluate the tag sequence for each position with greater contextual awareness and typically achieve higher accuracy compared to HMMs.

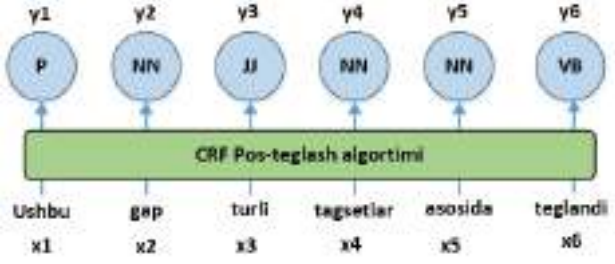

Fig. 2. A simplified structure of the linear-chain Conditional Random Field (CRF) model

In this Fig 2, the nodes $\boldsymbol{x_1, x_2, \ldots, x_n}$ in the lower layer represent the sequence of input words, while the nodes $\boldsymbol{y_1, y_2, \ldots, y_n}$ in the upper layer represent the corresponding sequence of tags. In a linear-chain CRF, the output (tag) nodes are interconnected via undirected edges, allowing them to jointly determine the optimal tag sequence by considering the influence of neighboring tags. There are no direct connections between the input nodes; all influences from the input sequence are modeled through the tag nodes. Thus, the CRF model combines both the dependencies between adjacent tags and the rich feature set of each word position (e.g., its meaning, morphological affixes, and surrounding words) to achieve accurate tagging.

In the CRF model, dynamic programming (e.g., the Viterbi or Forward-Backward algorithm) is also used to find the most probable sequence of tags. At each position, a score is computed for each tag. This score is derived from the transition from the previous tag to the current one and the sum of the characteristic features of the current tag at that position, weighted by the corresponding $\boldsymbol{\lambda}$ parameters. The necessary formulas are implemented in a recursive manner similar to that in HMM, except that instead of using predefined probabilities like $\boldsymbol{P(t_i|t_{i-1})}$ and $\boldsymbol{P(w_i|t_i)}$, scores of the form $\boldsymbol{exp\left(\sum_j \lambda_j f_j(t_{i-1}, t_i, W, i)\right)}$ are used. Below is a simplified pseudocode of the algorithm for finding the optimal tag sequence in the CRF model (using the Viterbi method).

Algorithm 2: Finding the Most Likely Tag Sequence in the CRF Model

 # Input: $X = [w_1, w_2, \ldots, w_n]$ *– sequence of words*
 # Output: $Y = [y_1, y_2, \ldots, y_n]$ *– most probable sequence of tags*
1 *# 1. Birinchi so‘z uchun boshlang‘ich ballar:*
2 **for** *each possible tag* **t**:
3 $score[1][t] = \sum_j \lambda_j * f_j(< Start >, t, X, 1)$
4 *# (transition from start state to t + features for* w_1*)*
5 $backpointer[1][t] = < Start >$
6 *# 2. Sequential dynamic programming:*
7 **for** i *from 2 to n*:
8 **for** *each tag* **t**:
9 $max_{score} = -\infty$
10 $best_{prev_{tag}}$ = None
11 **for** *each tag* $\boldsymbol{t_{prev}}$:
12 $s = score[i-1][t_{prev}] + \sum_j \lambda_j * f_j(t_{prev}, t, X, i)$
13 **if** $s > max_{score}$:
14 $max_{score} = s$
15 $best_{prev_{tag}} = t_{prev}$
16 $score[i][t] = max_{score}$
17 $backpointer[i][t] = best_{prev_{tag}}$
18 *# 3. Find the best final tag* y_n*:*
19 $best_{score} = -\infty$
20 $best_{last_{tag}} = None$
21 **for** *each tag* **t**:
22 **if** $score[n][t] > best_{score}$:
23 $best_{score} = score[n][t]$
24 $best_{last_{tag}} = t$
25 *# 4. Backtrace the full tag sequence::*

26 $y[n] = best_{last\ tag}$
27 **for** i *from n down to 2*:
28 $y[i-1] = backpointer[i][y[i]]$
29 **return** Y

In the above algorithm, among the features $\boldsymbol{f_j(t_{i-1}, t_i, X, i)}$, there may be rule-based features such as "tag t_i = NOUN ∧ word w_i ends with the suffix -ni." In classical CRF models, such features are designed manually by linguistic experts and extracted from the corpus. A specific characteristic of the Uzbek language is that suffixes at the end of a word play a significant role in determining its grammatical category. For instance, adding the suffix "-gan" to a verb can derive an adjective (past participle), or adding "-lik" to an adjective can derive a noun. Therefore, incorporating features based on such affixes into the CRF model improves its accuracy.

BiLSTM-CRF Neural Network Model

In recent years, deep learning methods have achieved significant success in natural language processing. In particular, Bidirectional Long Short-Term Memory networks (BiLSTMs) are capable of learning long-range dependencies within sequences, enabling more linguistically informed modeling of text. The BiLSTM-CRF architecture combines a traditional Conditional Random Field (CRF) model with a neural network, where the CRF layer makes tagging decisions based on rich representations (embeddings) generated for each word using both its left and right context[8].

The structure of the BiLSTM-CRF model can be described as follows: first, the input words are represented as numerical vectors[9]. These vectors are passed through two separate LSTM networks—one processes the sentence from the beginning to the end (left-to-right), while the other processes it in reverse (right-to-left)[10]. As a result, for each word position, two hidden state vectors are obtained that capture the left and right context: $\boldsymbol{h_i^{\rightarrow}}$ (left context) and $\boldsymbol{h_i^{\leftarrow}}$ (right context). These are concatenated to form a complete contextual representation vector $\boldsymbol{c_i} = [\boldsymbol{h_i^{\rightarrow}}; \boldsymbol{h_i^{\leftarrow}}]$.

Subsequently, instead of using a standard output layer (such as softmax or logistic regression) to predict tag probabilities for each $\boldsymbol{c_i}$, a CRF layer is applied. The CRF layer takes into account the dependencies and compatibility constraints between tags across the sequence and selects the most suitable tag sequence for the entire sentence.

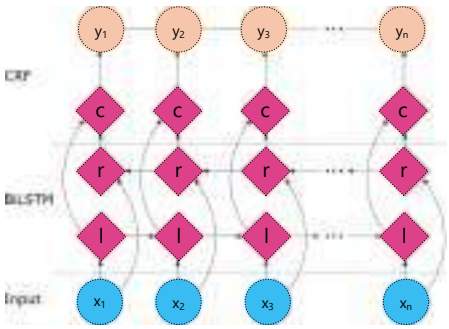

Fig. 3. BiLSTM-CRF Architecture

At the bottom layer, the input words $\boldsymbol{x_1, x_2, \dots, x_n}$ are fed into the neural network as concatenated word embeddings. In the middle layer, a bidirectional LSTM is employed: the leftward LSTM (l nodes) processes the sequence from left to right, while the rightward LSTM (r nodes) processes it from right to left. These two streams gather distinct contextual information for each position. Then, the outputs from both directions are concatenated to produce contextual representation vectors $\boldsymbol{c_1, c_2, \dots, c_n}$ at the top layer. Finally, the vectors $\boldsymbol{c_i}$ are passed to the CRF layer, which finds the optimal tag sequence $\boldsymbol{y_1, y_2, \dots, y_n}$. The strength of the BiLSTM-CRF model lies in its ability to simultaneously capture both the full-sentence context (i.e., signals from left and right neighbors and long-range dependencies) and the global consistency of the observed tag sequence. This makes BiLSTM-CRF particularly effective for long sentences or complex word pairs, where it often outperforms a standard CRF. While a plain CRF selects the current tag based solely on the previous tag, BiLSTM-CRF benefits from contextual signals from both past and future words across the entire sentence.

When applying the BiLSTM-CRF architecture to the Uzbek language, an additional important aspect must be considered: morphological complexity. The internal structure of words – stems and affixes – has a strong impact on meaning and can greatly aid in tagging. In the BiLSTM-CRF model, word representations typically incorporate not only pre-trained word embeddings, but also character- or syllable-level embeddings (e.g., whether a word starts with a capital letter, its final characters, syllabic structure, etc.).

BERT-CRF model

In contemporary NLP research, the success of Transformer architectures and their associated pre-trained models (such as BERT, RoBERTa, XLM-RoBERTa, etc.) has become increasingly evident. The BERT (Bidirectional Encoder Representations from Transformers) model is capable of producing contextual embeddings that capture the internal linguistic characteristics of a language through self-supervised learning on large-scale corpora.

BERT's bidirectional transformer base analyzes input text with full context, generating powerful, unified hidden representations (embeddings) for each word – or more precisely, for each subword token [6]. Therefore, fine-tuning a pre-trained BERT model for specific tasks has been demonstrated to be one of the most effective solutions, especially for low-resource language[11].

In the task of POS tagging, BERT is typically used in the following way: a pre-trained BERT model is fine-tuned on annotated Uzbek text data. On top of the BERT outputs, a tagging layer is added that predicts POS tags for each token. In the simplest setup, this prediction layer consists of independent softmax classifiers for each position.

The BERT-CRF architecture is similar to BiLSTM-CRF, except that transformer encoder layers are used instead of LSTM to model the context[12]. The BERT model is capable of splitting even unknown words in Uzbek (using WordPiece tokenization) and determining their meaning based on the surrounding context. This is particularly useful for the agglutinative nature of the Uzbek language, where new words are frequently formed through affixation. For example, the word "bormaganlardanmisiz" is divided into several subwords in the BERT model, and each part is analyzed within its context; due to the model's attention mechanism, long-distance dependencies are also taken into account. As a result, BERT can correctly interpret grammatical information within such complex word structures (e.g., verb + negation -magan + plural -lar + ablative -dan + question particle -mi +

respectful pronoun siz) and assign the appropriate tag to the word. Considering that traditional rule-based taggers struggle in such cases, the BERT model has a significant advantage in terms of contextual sensitivity and morphological awareness.

Several BERT models for the Uzbek language have recently been developed: UzBERT (Mansurov and Mansurov, 2021) – trained on 142 million words of clean text, and TahrirchiBERT (Mamasaidov and Shopulatov, 2023) – trained on 5 billion tokens of relatively "shovqinli" text (blogs, OCR books). Additionally, the multilingual mBERT (Devlin et al., 2019) is also available[13]. These models have so far been evaluated only using MLMA (Masked Language Modeling), but recently Bobojonova et al. (2025) tested these models on the POS tagging task[7]. According to their BBPOS study, fine-tuning specialized Uzbek BERT models for POS tagging achieved an average accuracy of 91%, which is significantly higher than both mBERT (multilingual) and rule-based taggers[14].

Experiments and Results

Dataset and Experimental Setup

Open-source annotated corpora in the Uzbek language are very limited. For this study, we used manually POS-tagged datasets consisting of 17038, 56616, and 77821 sentences. The composition of these three datasets is as follows.

TABLE I. COMPOSITION OF MANUALLY POS-TAGGED DATASETS IN CONLL-U FORMAT

№	POS tag	Dataset 1 (17038 sentences)	Dataset 2 (56616 sentences)	Dataset 3 (77821 sentences)
1.	N	3594 (18,92%)	37545 (24,5%)	49368 (32,21%)
2.	VB	3104 (16,34%)	33393 (21,79%)	45627 (29,77%)
3.	JJ	875 (4,61%)	10134 (6,61%)	12907 (8,42%)
4.	NUM	214 (1,13%)	2187 (1,43%)	2967 (1,94%)
5.	RR	755 (3,97%)	7523 (4,91%)	10283 (6,71%)
6.	P	896 (4,72%)	9432 (6,15%)	12844 (8,38%)
7.	II	459 (2,42%)	4564 (2,98%)	6206 (4,05%)
8.	C	246 (1,3%)	3706 (2,42%)	4771 (3,11%)
9.	Prt	0 (0%)	1813 (1,18%)	2581 (1,68%)
10.	MD	13 (0,07%)	1888 (1,23%)	2539 (1,66%)
11.	IM	36 (0,19%)	39 (0,03%)	55 (0,04%)
12.	UH	718 (3,78%)	329 (0,21%)	494 (0,32%)
13.	NER	1603 (8,44%)	8567 (5,59%)	11386 (7,43%)
14.	IB	2889 (15,21%)	5003 (3,26%)	6106 (3,98%)
15.	PUNCT	3594 (18,92%)	27151 (17,71%)	37725 (24,61%)
Total		18996	153274	205859

The data is presented in CONLL-U format, where each word appears on a separate line with its corresponding tag (in accordance with the Universal Dependencies standard, comprising 15 categories). The corpus mainly includes sentences from formal texts (news) and informal texts (literary works), and the tag distribution in the dataset is provided in Table I above. Categories such as IM (interjection), Prt (particle), UH (exclamation), and MD (modal verb) occur very infrequently – less than 2% of the total.

Training parameters for neural models were as follows: the BiLSTM-CRF model used a word **embedding size of 100** and a **hidden state size of 128** (for each direction). The **optimizer was Adam, with a learning rate of $5e^{-3}$**, and training was conducted over **20 epochs**. For the BERT-CRF model, the pre-trained multilingual BERT (mBERT, 110M parameters) was used. A single CRF layer was added on top of the output embeddings from its 12 transformer layers, and the entire model **was fine-tuned** for 3 epochs (learning rate $2e^{-5}$, batch size 16).

Analysis of Model Results

Accuracy and F1-score (overall micro-average) were used as evaluation metrics. **Accuracy** represents the proportion of correctly tagged tokens, while the **F1-score** reflects the harmonic mean of precision and recall based on true/false positives per tag. In this task, accuracy is generally sufficient, since each token belongs to a single class. Below are the **average accuracy (%)** and **F1-score (%)** results on the test set for different models:

TABLE II. POS TAGGING RESULTS OF MODELS FOR 3 UZBEK DATASETS (ACCURACY AND F1 ON TEST SET, IN PERCENT)

Model	Dataset 1		Dataset 2		Dataset 3	
	Accuracy (%)	*F1 (%)*	*Accuracy (%)*	*F1 (%)*	*Accuracy (%)*	*F1 (%)*
HMM	82.0	81.1	83.1	83.5	84.1	84.5
CRF	84.7	84.5	86.3	86.9	88.3	87.5
BiLSTM-CRF	86.2	86.4	89.6	90.1	91.0	90.6
BERT-CRF	88.6	89.1	92.4	92.6	93.1	93.4

As seen from Table II above, the statistical model HMM achieved the lowest performance on the **1st dataset containing 17038 sentences** (~82% accuracy). The CRF model, due to additional features and discriminative training, significantly outperformed HMM with ~85% accuracy. Deep learning models performed even better: BiLSTM-CRF reached ~86% accuracy, while BERT-CRF achieved ~89%. As expected, the BERT-CRF model showed the highest result, which is comparable to the level seen in some high-resource languages.

For the 3rd dataset consisting of 77821 sentences, the HMM model achieved ~84% accuracy, the CRF model ~86%, the BiLSTM-CRF model ~91%, and the BERT-CRF model ~93%. As expected, the BERT-CRF model demonstrated the highest result. Notably, Bobojonova et al. (2025) reported 91% accuracy for the BERT model on their smaller corpus consisting of 500 sentences. In our larger corpus of 77821 sentences, the BERT-based POS tagger achieved 93% accuracy, indicating that the model was trained more robustly as the dataset size increased.

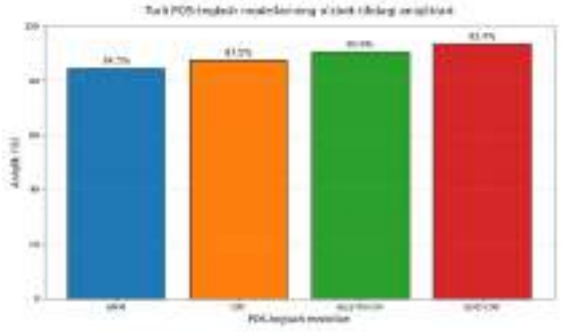

Fig. 4. Comparison of tagging accuracies of different POS tagging models for the Uzbek language

A more detailed analysis of the results shows that most of the HMM model's errors were related to new or uncommon

words. In cases where words had not been seen during training, the HMM model assigned zero probability and was forced to choose an incorrect tag. For example, the word *"badiiy"*, which did not appear in the training data, was incorrectly tagged as an adverb by HMM in a test sentence where it preceded a noun (while it should have been tagged as an adjective). Such cases occurred less frequently in the CRF model, because it had already been *"aytilgan"* that the feature ends with *"-iy"* is typical of adjectives, and CRF correctly tagged *"badiiy"* as an adjective based on this clue. Thus, it can be concluded that CRF's ability to use manually defined linguistic features plays an important role in the Uzbek language.

At the same time, some limitations of CRF were also observed: for example, **punctuation marks** such as *commas* and *periods* were sometimes incorrectly tagged as a different category. This could be explained by the fact that the feature set did not provide sufficient distinguishing characteristics for punctuation. Since in the training examples the tag for commas and periods was always **"PUNCT"** the model failed to learn their contextual usage and occasionally made mistakes.

In the BiLSTM-CRF model, errors were further reduced. This model tends to perform correct tagging even in complex contexts. For example, in the sentence *"U olmani bizga berdi"* ("He gave us the apple"), the word *"olma"* can have two meanings: *olma* (fruit, noun) or *ol+ma* (action, verb). HMM and CRF models tagged this word as a verb based on its more frequent occurrence, whereas in the actual meaning it should have been a noun. The BiLSTM-CRF model, however, correctly tagged *"olma"* as a noun by taking into account the word order in the sentence and the suffix *"ni"*. Thus, the neural network understands the context more deeply and outperforms traditional models in distinguishing polysemous words. In the BERT-CRF model, almost all such cases were correctly resolved. Even in more complex sentences, BERT more accurately distinguished homonyms. For instance, in the sentence *"Sayyora haqida gapiring"*, the word *"sayyora"* was understood by the BERT model to be a noun (meaning planet), not a proper name (NER), based on the context (*"haqida gapiring"*), and it was tagged as a noun. Some simple models, however, incorrectly tagged it as NER because it started with a capital letter.

The above-mentioned models failed to learn certain rarely occurring tags. For example, **interjections (UH)** were incorrectly redirected to other tags by HMM and CRF due to their very low frequency in the corpus. Even in the BiLSTM and BERT models, the F1 score for these tags was low (around ~50%) because there were not enough instances during training. This shows that even the most advanced model cannot perform well on cases that are completely absent or very rare in the data. This issue can only be resolved by increasing the size of the corpus.

CONCLUSION

In conclusion, this study analyzed and comparatively evaluated various statistical and neural models for part-of-speech (POS) tagging in the Uzbek language. The models HMM, CRF, BiLSTM-CRF, and BERT-CRF were tested on three datasets consisting of 17038, 56616, and 77821 sentences, respectively. It was demonstrated that classical statistical models such as HMM and CRF, based on probabilistic concepts in linguistics, can achieve a certain level of success. In particular, the CRF model outperformed HMM in accuracy due to its ability to incorporate contextual and feature-based information. Meanwhile, deep learning-based models such as BiLSTM-CRF and BERT-CRF proved capable of achieving state-of-the-art results in Uzbek POS-tagging tasks. The BiLSTM-CRF model demonstrated strong performance in resolving ambiguous cases thanks to its effective utilization of contextual information, while the BERT-CRF model, leveraging knowledge from pre-trained language models, achieved very high accuracy (approximately 93%) on the third dataset. This, in turn, illustrates the effectiveness of modern approaches for low-resource languages: if large-scale open text data is available for a language, pre-training transformer models on them and then fine-tuning on a smaller annotated dataset is the most effective method for achieving top performance..

REFERENCES

[1] B. Elov, & N. Xudayberganov, (2024). O'zbek tili korpusi matnlarini pos teglash usullari. Computer Linguistics: problems, solutions, prospects, 1(1).

[2] https://domino.ai/blog/named-entity-recognition-ner-challenges-and-model

[3] N. Xuan Bach, T. Khuong Duy, & T. Minh Phuong, (2019). A POS tagging model for Vietnamese social media text using BiLSTM-CRF with rich features. In PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III 16 (pp. 206-219). Springer International Publishing.

[4] A. Murat, & S. Ali, (2024). Low-resource POS tagging with deep affix representation and multi-head attention. IEEE Access.

[5] L. Bobojonova, A. Akhundjanova, P. Ostheimer, & S. Fellenz, (2025). BBPOS: BERT-based Part-of-Speech Tagging for Uzbek. arXiv preprint arXiv:2501.10107.

[6] B. E. Boltayevich, S. S. Samariddinovich, S. K. Mirdjonovna, E. Adalı, & Z. Y. Yuldashevna, (2023, September). POS taging of Uzbek text using hidden markov model. In 2023 8th International Conference on Computer Science and Engineering (UBMK) (pp. 63-68). IEEE.

[7] A. Bărbulescu, & D. Morariu, (2020). Part of Speech Tagging Using Hidden Markov Models. International Journal of Advanced Statistics and IT&C for Economics and Life Sciences, 10(1).

[8] R. Hoojon, & A. Nath, (2023, March). BiLSTM with CRF Part-of-Speech Tagging for Khasi language. In 2023 4th International Conference on Computing and Communication Systems (I3CS) (pp. 1-7). IEEE.

[9] S. Arslan, (2024). Application of BiLSTM-CRF model with different embeddings for product name extraction in unstructured Turkish text. Neural Computing and Applications, 36(15), 8371-8382.

[10] J. Liu, C. Sun, & Y. Yuan, (2020, November). The BERT-BiLSTM-CRF question event information extraction method. In 2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT) (pp. 729-733). IEEE.

[11] Z. Z. Hlaing, K. Y. Thu, T. Supnithi, & P. Netisopakul, (2022). Improving neural machine translation with POS-tag features for low-resource language pairs. Heliyon, 8(8).

[12] L. Zhang, & Y. Li. Finding Product Problems from Online Reviews Based on BERT-CRF Model.

[13] J. Devlin, W. M. Chang, K. Lee, & K. Toutanova, (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).

[14] D. Li, Y. Tu, X. Zhou, Y. Zhang, & Z. Ma, (2022). End-to-end chinese entity recognition based on bert-bilstm-att-crf. ZTE Communications, 20(S1), 27