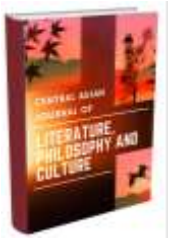

CENTRAL ASIAN JOURNAL OF LITERATURE, PHILOSOPHY, AND CULTURE

https://cajlpc.casjournal.org/index.php/CAJLPC

Volume: 07 Issue: 02 | April 2026 ISSN: 2660-6828

Article

State-of-the-Art NER Models for the Uzbek Language

Madina Samatboyeva

1. Department of Computational Linguistics and Digital Technology. Tashkent State University of Uzbek Language and Literature, Tashkent, Uzbekistan

* Correspondence: msamatboyeva@gmail.com

Abstract: Named Entity Recognition (NER) is one of the fundamental tasks in Natural Language Processing and plays an important role in many applications such as information extraction, machine translation, and question answering systems. In recent years, machine learning and deep learning approaches have significantly improved the performance of NER systems. However, developing accurate NER models for low-resource languages such as Uzbek remains a challenging task due to the limited availability of annotated corpora and linguistic resources. This paper reviews state-of-the-art NER models used in machine learning for the Uzbek language, including traditional statistical methods and modern neural network architectures. In particular, models based on Conditional Random Fields, Bidirectional LSTM, and transformer-based architectures such as BERT and XLM-RoBERTa are analyzed. The study discusses their effectiveness, advantages, and limitations in the context of Uzbek language processing. The findings highlight that transformer-based multilingual models demonstrate the best performance for Uzbek NER tasks and provide promising directions for future research.

Keywords: Named Entity Recognition, Uzbek language, machine learning, deep learning, BERT, XLM-RoBERTa, Bidirectional LSTM, Natural Language Processing, corpus linguistics

Citation: Samatboyeva, M. State-of-the-Art NER Models for the Uzbek Language. Central Asian Journal of Literature, Philosophy, and Culture 2026, 7(2), 147-157

Received: 10th Dec 2025
Revised: 21th Jan 2026
Accepted: 04th Feb 2026
Published: 26th Mar 2026

1. Introduction

In recent years, the rapid development of artificial intelligence technologies has significantly influenced the field of Natural Language Processing. One of the most important tasks in this field is Named Entity Recognition (NER), which focuses on identifying and classifying entities such as person names, organizations, locations, dates, and numerical expressions in textual data. NER plays a crucial role in many NLP applications, including information extraction, text summarization, question answering systems, and machine translation. Traditionally, NER systems relied on rule-based approaches and statistical models. Early studies widely applied methods such as Hidden Markov Model and Conditional Random Fields for sequence labeling tasks. Although these approaches achieved reasonable results, they required extensive manual feature engineering and linguistic resources [1, 2].

With the emergence of DL (deep learning) techniques, neural network architectures began to dominate NER research. Models based on Bidirectional LSTM demonstrated significant improvements by capturing contextual information from both directions of a sentence. More recently, transformer-based architectures such as BERT and XLM-RoBERTa have achieved state-of-the-art results in many NLP tasks, including NER [3, 4].

Despite these advancements, developing accurate NER systems for low-resource languages remains challenging. The Uzbek language, which belongs to the Turkic language family, has relatively limited annotated corpora and linguistic resources for

training machine learning models. This limitation makes it difficult to achieve high-performance NER systems compared to resource-rich languages such as English or Chinese. Therefore, this study aims to review and analyze state-of-the-art machine learning models for Named Entity Recognition in the Uzbek language. The paper discusses the advantages and limitations of different NER approaches and highlights the potential of transformer-based multilingual models for improving NER performance in Uzbek text processing [5, 6].

Literature review

In recent years, several researchers have made significant contributions to the field of Named Entity Recognition (NER) by developing state-of-the-art models and techniques. Wang et al. (2025) [7] proposed a novel approach called GPT NER, which leverages large language models (LLMs) to transform traditional sequence labeling tasks into generation tasks. This method demonstrated improved performance, particularly in low resource settings where annotated data is limited. Zhang et al. (2025) [8] introduced KoGNER, a framework that incorporates knowledge graph integration into NER models, enabling enhanced contextual representations and superior performance across various domains. Another important contribution was made by Zhou et al. (2023) [9], who presented UniversalNER, a model that utilizes distillation from large language models to create more efficient multilingual NER systems. In 2024, Mayhew et al. [10] developed a multilingual benchmark called Universal NER, providing standardized evaluations across many languages and highlighting the strengths and weaknesses of different multilingual NER approaches. Finally, Zhang et al. (2025) [11] proposed a hybrid model called RoBERTa NER for ancient language NER tasks, combining transformers with classical sequence labeling techniques to achieve competitive results. These studies collectively showcase the rapid progress and innovation in NER research, emphasizing multilingual and large model based approaches.

In the process of automatic NER identification for the Uzbek language, conventional models such as CRF and HMM were dominant in the early 2000s, later progressing to hybrid models combining LSTM, BiLSTM and CRF, and most recently advancing to transformer based models such as BERT, RoBERTa, and XLM R [12].

2. Materials and Method

A comprehensive and systematic analysis of the existing models dealing with Named Entity Recognition (NER) for the Uzbek language is provided in this study. It interacts qualitative analysis, comparative and model-oriented review techniques as methods of research.

To approach this, we first employed a systematic literature review method that enabled us to identify and analyze developments in NER over the past years, with specific attention on approaches such as machine learning, deep learning and transformer-based methods. Studies were chosen based on their scientific merit, recent publication (2015–2025) and applicability to low-resource languages. This survey covers classic models such as Conditional Random Fields (CRF) and modern neural architectures like BiLSTM-CRF, BERT, XLM-RoBERTa.

Second, the performance and characteristics of various NER models were evaluated through a comparative analysis method. Models compared on a few criteria:

ability to capture contextual information,
dependency modeling between labels,
computational complexity,
adaptability to low-resource languages,
in the performance of Uzbek language processing tasks.

Third, the research employs a frequency-based categorization methodology to sort NER systems into three major classification groups:

statistical machine learning models (CRF, HMM, etc.)
neural network-based models (BiLSTM/BiLSTM-CRF, etc.),
transformer-based models (e.g., BERT, XLM-RoBERTa).

This classification facilitates a systematic approach to understanding the evolution of NER technologies, illustrating the shift from feature-based models towards context-aware deep learning architectures.

Moreover, a descriptive-analytical approach was applied to study the internal mechanisms of some models. For example, the feature engineering and sequence labeling aspects of CRF models were examined; the bidirectional context learning and label dependency optimization potential of BiLSTM-CRF architectures was assessed; while transformer models were unveiled for their attention mechanisms or pretrained representations.

In addition, they include an example-based analysis with Uzbek native data to show how various models behave in NER tasks. This method helps highlight the strengths and weaknesses of each model with regards to morphologically rich and low-resource language data.

At the final stage, a synthesis method was used to integrate results of all analyzed models and find out what methods performed best for Uzbek NER. These results highlight the potential of transformer-based multilingual architectures, while simultaneously re-establishing that hybrid models such as BiLSTM-CRF are still useful in certain situations.

3. Result and Discussion

Effective NER models can be divided into three types: 1**) Statistical machine learning models; 2) neural network–based approaches (DL); 3) transformer-based models "Fig.1".**

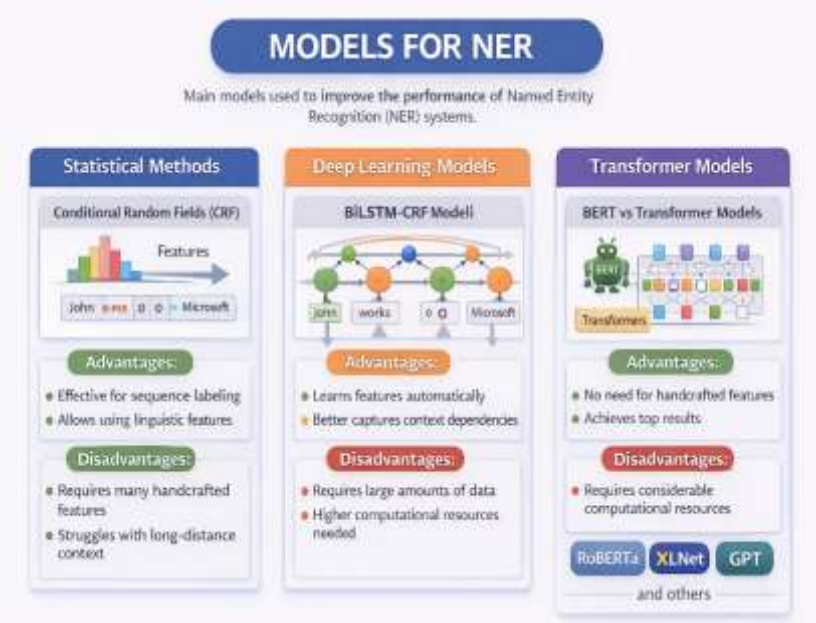

Figure 1. Best models for NER

Conditional Random Fields (CRF)

Conditional Random Fields (CRF) is a probabilistic sequence labeling model widely used in Named Entity Recognition to assign context-aware labels to words, leveraging handcrafted linguistic features while effectively capturing local dependencies, though it requires extensive feature engineering and struggles with long-range contextual information. Conditional Random Fields (CRF) is a probabilistic model used for sequence data, and in tasks like Named Entity Recognition (NER), its purpose is to assign a label to each word while taking into account not only the word itself but also its surrounding context [13].

For a CRF model to function, features are extracted for each word, which help the model understand the word and its context. Common features include:

1. **The word itself (word)** – the original form of the text
2. **Capitalization (istitle, isupper)** – helps identify personal names or organization names
3. **Numbers or special characters (isdigit, punctuation)**
4. **Prefixes and suffixes (-soft, -bank)** – useful for recognizing company names
5. **Previous and next words** – to capture the surrounding context

CRF sequence diagram is a visual representation used in Natural Language Processing to illustrate how sequence labeling is performed. It shows the relationship between words in a sentence and their corresponding labels. In this diagram, each token is connected to a possible tag, and arrows indicate transitions between tags across the sequence. Unlike simple classifiers, the CRF model considers the context of the entire sentence when assigning labels "Fig.2.".

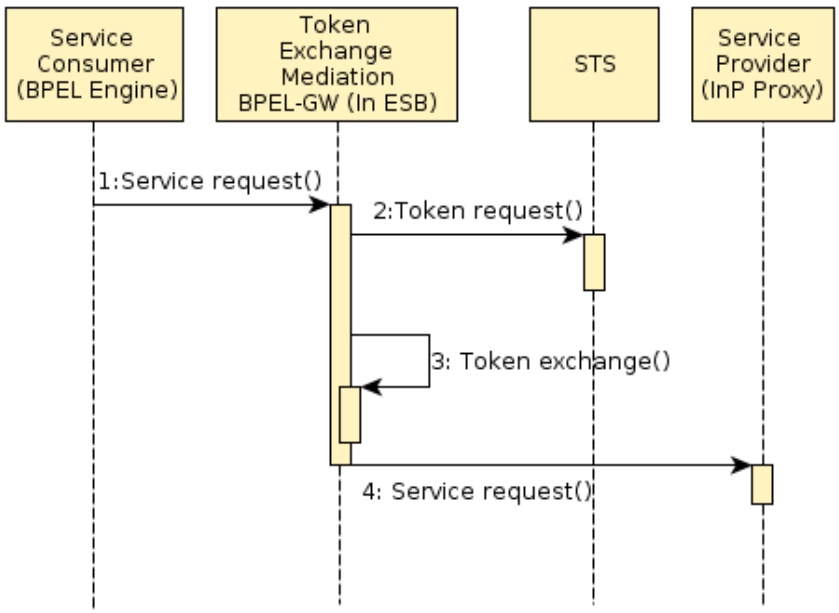

Figure 2. CRF Sequence Labeling Diagram

This allows the model to capture dependencies between neighboring tags. CRF sequence diagrams are commonly used in tasks such as Named Entity Recognition, part-of-speech tagging, and other structured prediction problems, helping researchers understand how the model determines the most probable label sequence.

Table I. CRF Model Demonstration In An Uzbek Language Example

So'z	*prev_word*	*next_word*	*istitle*	*isupper*	*label*
Alisher	BOS	hisobchi	True	False	PER
hisobchi	Alisher	,	False	False	O
,	hisobchi	Xalqbankda	False	False	O
Eurobankda	,	ishlaydi	True	False	ORG
ishlaydi	Xalqbankda	.	False	False	O

BOS – indicates the beginning of a sentence.

EOS – indicates the end of a sentence.

PER O O ORG O O
| | | | | |
Alisher → hisobchi → , → Xalqbankda → ishlaydi → .

CRF differs from simple classifiers in that it takes into account the dependencies between words.

For example, the *PER* (person) tag usually appears consecutively with *O* or another PER tag. In the example above, **"Alisher"** is a person's name; it could have been followed by a surname, e.g., **"Alisher Hamidov."** Or it could appear next to an outside word, as it did with the word **"accountant."** The *ORG* tag, on the other hand, can be associated with a preceding O tag or a comma "Fig 3.".

In a CRF model, the probabilities of all possible tags are calculated for each word, and the most optimal sequence is selected. This process is performed using the Viterbi algorithm [14].

As a result, the sequence of tags with the highest probability for the sentence. The CRF model considers the dependencies in the sequence and outputs accurate NER tags.

```python
import sklearn_crfsuite

# Sentence and labels
sentence = [("Alisher", "NNP"), ("hisobchi", "NN"), (",", ","), ("Eurobankda", "NNP"), ("ishlaydi", "VBZ"), (".", ".")]
labels = ["PER", "O", "O", "ORG", "O", "O"]

# Function to extract features for each word
def word2features(sent, i):
    word = sent[i][0]
    features = {
        'word.lower()': word.lower(),       # lowercase form of the word
        'word.istitle()': word.istitle(),   # whether the word starts with a capital letter
        'word.isupper()': word.isupper(),   # whether the word is all uppercase
    }
    if i > 0:
        features['prev_word'] = sent[i-1][0].lower()  # previous word
    else:
        features['BOS'] = True  # Beginning of sentence
    if i < len(sent)-1:
        features['next_word'] = sent[i+1][0].lower()  # next word
    else:
        features['EOS'] = True  # End of sentence
    return features

# Function to convert entire sentence into feature list
def sent2features(sent):
    return [word2features(sent, i) for i in range(len(sent))]

# Prepare training data
X_train = [sent2features(sentence)]
y_train = [labels]

# Create CRF model
crf = sklearn_crfsuite.CRF(
    algorithm='lbfgs',            # optimization algorithm
    max_iterations=100,           # maximum number of iterations
    all_possible_transitions=True  # consider all possible label transitions
)

# Train the CRF model
crf.fit(X_train, y_train)

# Test on the same sentence
pred = crf.predict(X_train)
print(pred)  # Output: [['PER', 'O', 'O', 'ORG', 'O', 'O']]
```

Figure 3. CRF Example in Python

BiLSTM-CRF

The combination of *Bidirectional LSTM (BiLSTM)* and *Conditional Random Fields (CRF)* is a classical and highly effective architecture for Named Entity Recognition (NER). The BiLSTM component captures the context of words in a sentence in both directions, i.e., forward and backward. The CRF component, on the other hand, selects the most probable sequence of labels by considering the dependencies between them.

The BiLSTM-CRF model is primarily used for NER and other sequence labeling tasks. It consists of two main components:

1. BiLSTM (Bidirectional Long Short-Term Memory): Learns the context of words in the text in both directions.
2. CRF (Conditional Random Field): Optimizes the output label sequence by accounting for the logical dependencies between labels.

Table II. Similarities and Differences Between Bilstm and CRF

FEATURE	BILSTM	CRF	EXPLANATION
MAIN PURPOSE	Learn the context in sequences and generate a vector for each word	Optimize the sequence of labels and enforce logical constraints	BiLSTM captures internal context, CRF models label dependencies
INPUT	Word vectors (embeddings)	Output from BiLSTM (score vectors)	CRF evaluates the BiLSTM outputs
OUTPUT	Context-aware vector for each word	Sequence of labels	CRF output is the final predicted labels
CONTEXT HANDLING	Forward and backward (bidirectional)	Dependencies between labels (transition matrix)	BiLSTM captures semantic context, CRF captures grammatical dependencies
SEQUENCE OPTIMIZATION	No (words are scored independently)	Yes (finds global optimum using Viterbi algorithm)	CRF preserves sequential label constraints
MATHEMATICAL MODEL	LSTM equations with tanh, sigmoid, forget gate	Conditional Random Field, log-linear model	Two different models, can be combined for sequence labeling
ADVANTAGES	Captures long-range context, bidirectional	Produces logically consistent label sequences, global optimum	BiLSTM + CRF forms a strong combination
DISADVANTAGES	Does not consider label dependencies	Does not capture semantic context of words	Hence, BiLSTM and CRF are often used together

LSTM (Long Short-Term Memory) learns long-term dependencies in sequences, and Bidirectional LSTM captures context in both directions: forward LSTM reads left-to-right, backward LSTM reads right-to-left, producing a context-aware vector for each word "Fig.4.".

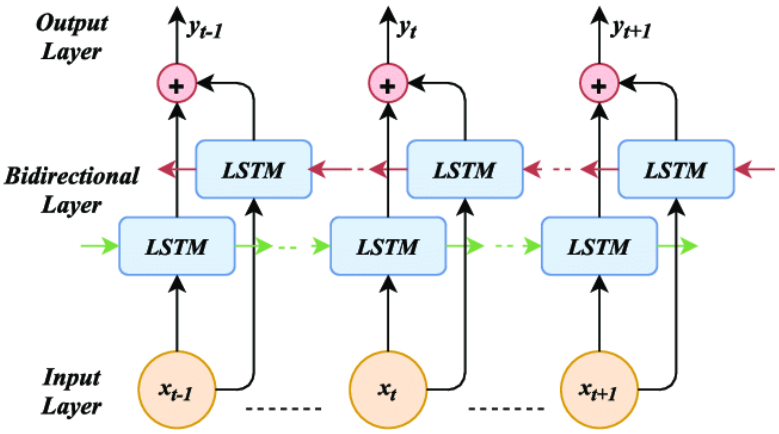

Figure 4. BiLSTM (Bidirectional LSTM) model

The vectors output by the BiLSTM are passed to the CRF layer. The CRF predicts a label for each word and selects the sequence of labels based on a global optimum, enforcing rules such as preventing an "I-PER" tag from following an "O" tag. In summary, the first stage of NER – entity recognition – is performed using BiLSTM, and the second stage – classification or labeling – is handled by the CRF model. By combining these two powerful models, highly accurate results can be achieved. Words are the tokens in the input text, embeddings convert each word into a vector, BiLSTM learns context in both forward and backward directions, CRF logically optimizes the sequence of labels (e.g., preventing "I-PER" from following "O"), and the labels are the final NER output [15].

How BiLSTM Works (Step-by-Step)

1. Sentence Tokenization: *"Murod Samarqand viloyati Jomboy tumanida yashaydi."*
[Murod] [Samarqand] [viloyati] [Jomboy] [tumanida] [yashaydi]

2. Word Embedding
Murod → x1
Samarqand → x2
viloyati → x3
Jomboy → x4
tumanida → x5
yashaydi → x6

3. Forward LSTM (read from left to right)
Murod → Samarqand → viloyati → Jomboy → tumanida → yashaydi
h1_forward
h2_forward
h3_forward
h4_forward
h5_forward
h6_forward
h3_forward
= *Murod* + *Samarqand* + *viloyati* kontekstini biladi

4. Backward LSTM (read from right to left)
yashaydi → tumanida → Jomboy → viloyati → Samarqand → Murod
h1_backward
h2_backward
h3_backward
h4_backward
h5_backward
h6_backward
h4_backward
= *Jomboy* + *tumanida* + *yashaydi* kontekstini biladi

5. Combining the forward and backward results
h1 = [h1_forward ; h1_backward]
h2 = [h2_forward ; h2_backward]
h3 = [h3_forward ; h3_backward]
h4 = [h4_forward ; h4_backward]
h5 = [h5_forward ; h5_backward]
h6 = [h6_forward ; h6_backward]
As a result, each word understands the context of the entire sentence.
Named Entity Recognition (NER) results: The BiLSTM model assigns a tag to each word.

Murod → *PER*
Samarqand → *LOC*
viloyati → *LOC*
Jomboy → *LOC*
tumanida → *LOC*
yashaydi → *O*

- ***PER*** → *Person (inson)*
- ***LOC*** → *Location (joy nomi)*
- ***O*** → *Oddiy so'z*

To identify the word **"Jomboy,"** the model looks at both sides: *Left context:* Samarkand region; *right context:* lives in the district. Therefore, the model can more accurately understand that **"Jomboy"** is a **location name**.

BERT model

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model for natural language processing (NLP) that learns words in both directions (left-to-right and right-to-left) with context, unlike previous models (e.g., GPT) which used only one-way context, and it operates based on Transformer encoder layers.

BERT utilizes deep bidirectional transformers pre-trained on large text corpora, enabling models to capture contextual information from both directions, significantly improving performance on various language understanding tasks.

BERT is one of the most significant breakthroughs in natural language processing (NLP) based on deep learning, enabling major advancements in various NLP tasks. In this approach, the model simultaneously considers both the preceding and following context of words, which is crucial for accurately understanding the meaning of phrases. As emphasized by Devlin and colleagues, the bidirectional pre-training approach allows the model to learn deeper relationships between words, and this approach can effectively handle even very complex NLP tasks with just a simple additional layer. The strength of BERT lies in creating universal language representations, meaning it can be adapted to multiple NLP tasks. The success of BERT has also inspired numerous subsequent innovations, such as RoBERTa, ALBERT, and other transformer-based models.

To understand BERT architecture, let's consider an example in Uzbek:

"Asakabankda Jamshid bosh mutaxassis bo'lib ishlaydi."

The sentence is first tokenized**,** and then each token is converted into a vector**.** Next, the model is trained right-to-left and left-to-right, capturing context from both directions. As a result, each analyzed unit is tagged. The BERT model uses pretrained NER tags**.** BERT uses the WordPiece tokenizer, so long or uncommon words are split into subtokens "Fig.5.".

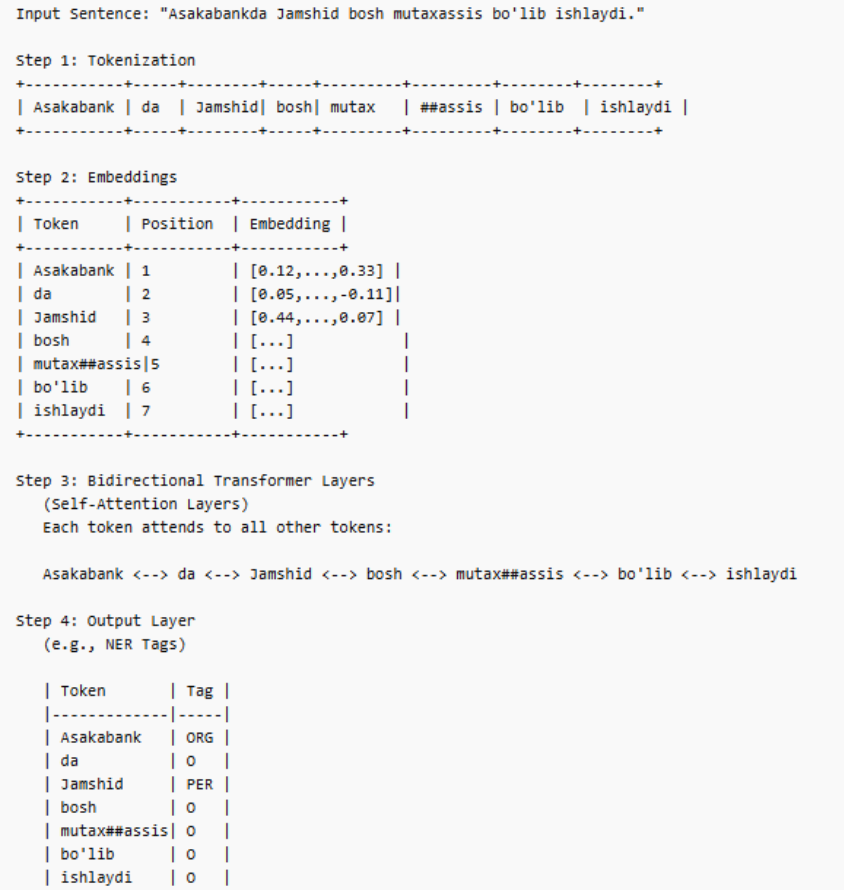

Figure 5. BERT Architecture

Each token has *three types* of embeddings: *Token Embedding* – the word itself, *Position Embedding* – its position in the sentence, and *Segment Embedding* – which sentence/segment it belongs to "Fig.6.".

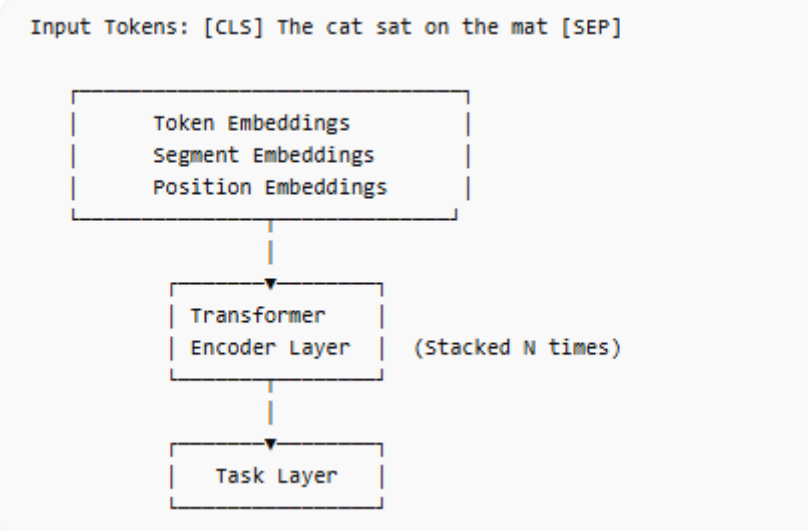

Figure 6. Embedding process in the BERT model

Token	Token emb	Position emb	Segment emb	Combined
Asakabank	[0.12..]	[0.01..]	[0.0..]	[0.13..]
Jamshid	[0.44..]	[0.03..]	[0.0..]	[0.47..]

Column / part	Explanation in english
Token	The actual word or subword from the input sentence. Example: "asakabank" or "jamshid".
Token emb	The **token embedding** – a vector representing the meaning of the token itself, independent of its position. Example: is a numerical vector encoding "asakabank".
Position emb	The **position embedding** – a vector that encodes the token's position in the sentence. Example: shows that "asakabank" is the 1st word in the sentence.
Segment emb	The **segment embedding** – a vector indicating which sentence or segment the token belongs to (useful for tasks with multiple sentences). Example: if it's in the first segment.
Combined	The final embedding used by bert for the token, obtained by **adding token, position, and segment embeddings** together. Example:

The BERT model simultaneously identifies NER entities and tags them, i.e., classifies them. Non-NER entities are labeled as "O" (outside) "Fig.7.".

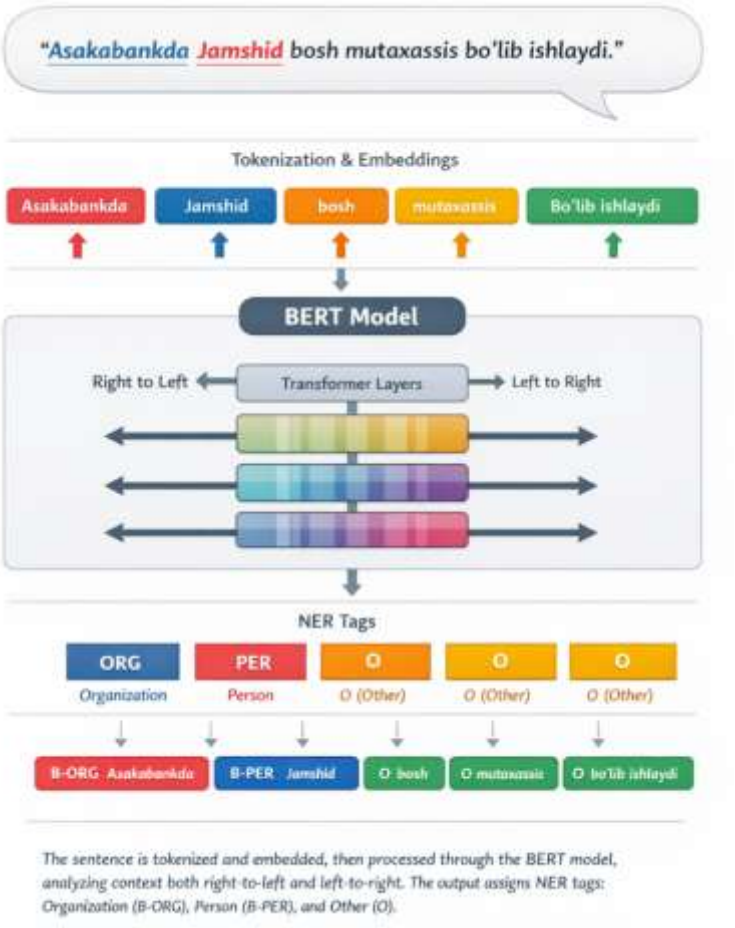

Figure 7. Tagging process in the BERT model

The strengths of the BERT model:

1. Bidirectional Context Understanding

- BERT reads text both left-to-right and right-to-left, capturing the full context of each word.
- This improves understanding of polysemous words (words with multiple meanings).

2. Pretrained on Large Corpora

- BERT is pretrained on massive datasets like BookCorpus and Wikipedia, giving it strong language understanding even before fine-tuning.

3. Fine-Tuning Flexibility

- Can be fine-tuned for various NLP tasks: NER, question answering, sentiment analysis, text classification, etc., with minimal task-specific architecture changes.

4. State-of-the-Art Performance

- Achieves high accuracy on benchmarks like GLUE, SQuAD, and CoNLL, often outperforming previous models.

5. Handles Long-Range Dependencies

- The Transformer architecture allows BERT to capture relationships between words that are far apart in a sentence.

6. Reusable Embeddings

- Output embeddings from BERT can be used as feature representations for downstream tasks without retraining from scratch.

7. Robust to Complex Sentences

- Effective in understanding complex sentence structures and nuanced language patterns.

4. Conclusion

This study provides a comprehensive overview of the current state-of-the-art Named Entity Recognition (NER) models applied to the Uzbek language, a low-resource language with limited annotated corpora. Traditional statistical methods such as Conditional Random Fields (CRF) have been widely used for sequence labeling tasks, offering simplicity and interpretability, but they often struggle with complex linguistic patterns. Neural network approaches, including Bidirectional LSTM (BiLSTM), improve context awareness by processing sequences in both directions, enhancing NER performance for morphologically rich languages like Uzbek. Transformer-based architectures, especially BERT and XLM-RoBERTa, leverage pretraining on large multilingual corpora, providing superior performance in capturing semantic and syntactic nuances. Comparative analysis demonstrates that multilingual transformers consistently outperform both statistical and recurrent neural models, particularly in handling noisy or domain-specific texts. Despite their effectiveness, challenges remain, including the scarcity of high-quality annotated Uzbek datasets and computational demands of large models. The findings indicate that adopting multilingual transformers, possibly combined with data augmentation or semi-supervised learning, offers the most promising direction for advancing Uzbek NER research, supporting applications such as information extraction, question answering, and machine translation.

REFERENCES

[1] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, G. Wang, and C. Guo, "GPT NER: Named entity recognition via large language models," in Findings of the Association for Computational Linguistics: NAACL 2025, 2025. [Online]. Available: https://aclanthology.org/2025.findings-naacl.239

[2] H. Zhang, W. Li, D. Huang, Y. Tang, Y. Chen, P. Payne, and F. Li, "KoGNER: A novel framework for knowledge graph integration on biomedical named entity recognition," 2025. [Online]. Available: https://arxiv.org/abs/2503.15737

[3] W. Zhou, S. Zhang, Y. Gu, M. Chen, and H. Poon, "UniversalNER: Targeted distillation from large language models for open named entity recognition," 2023. [Online]. Available: https://arxiv.org/abs/2308.03279

[4] S. Mayhew et al., "Universal NER: A gold-standard multilingual named entity recognition benchmark," in Proc. NAACL, 2024. [Online]. Available: https://aclanthology.org/2024.naacl-long.243

[5] Y. Zhang, M. Liu, H. Tang, S. Lu, and L. Xue, "Simple named entity recognition (NER) system with RoBERTa for ancient Chinese," in Proc. Second Workshop on Ancient Language Processing, 2025. [Online]. Available: https://aclanthology.org/2025.alp-1.27

[6] B. B. Elov and M. T. Samatboyeva, "Process of automatic NER identification in the Uzbek language corpus," in Proc. Int. Conf. Computational Linguistics and NLP, 2024.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423

[8] "BERT (language model)," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/BERT_(language_model)

[9] "BERT language model definition," TechTarget. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

[10] "Bidirectional encoder representations from transformers (BERT)," Emergent Mind. [Online]. Available: https://www.emergentmind.com/topics/bidirectional-encoder-representations-from-transformers-bert

[11] X. Li, X. Sun, Y. Sun, and J. Li, "Named entity recognition as dependency parsing," in Proc. ACL, 2020, pp. 6470–6480. doi: 10.18653/v1/2020.acl-main.577

[12] Z. Li, Z. Zhao, Y. Wei, and Y. Sun, "FLAT: Chinese NER using flat-lattice transformer," in Proc. ACL, 2020, pp. 6836–6842. doi: 10.18653/v1/2020.acl-main.611

[13] Y. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in Proc. NAACL-HLT, 2016, pp. 260–270. doi: 10.18653/v1/N16-1030

[14] G. Luo, X. Huang, C. Lin, and Z. Nie, "Joint entity recognition and disambiguation," in Proc. EMNLP, 2015, pp. 879–888. doi: 10.18653/v1/D15-1109

[15] A. Radford et al., "Language models are unsupervised multitask learners," OpenAI, Tech. Rep., 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf