UZNER: A Benchmark for Named Entity Recognition in Uzbek

Aizihaierjiang Yusufu[1], Liu Jiang[1], Abidan Ainiwaer[2], Chong Teng[1], Aizierguli Yusufu[3], Fei Li[1], and Donghong Ji[1](✉)

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
{azhar520A,tengchong,lifei_csnlp,dhji}@whu.edu.cn

[2] School of Information Management, Wuhan University, Wuhan, China

[3] School of Computer Science and Technology, Xinjiang Normal University, Urumqi, China
abida1020@whu.edu.cn

Abstract. Named entity recognition (NER) is a key task in natural language processing, and entity recognition can provide necessary semantic information for many downstream tasks. However, the performance of NER is often limited by the richness of language resources. For low-resource languages, NER usually performs poorly due to the lack of sufficient labeled data and pre-trained models. To address this issue, we manually constructed a large-scale, high-quality Uzbek NER corpus of Uzbek, and experimented with various NER methods. We improved state-of-the-art baseline models by introducing additional features and data translations. Data translation enables the model to learn richer syntactic structure and semantic information. Affix features provide knowledge at the morphological level and play an important role in identifying oversimplified low-frequency entity labels. Our data and models will be available to facilitate low-resource NER.

Keywords: Named Entity Recognition · Low resource · Uzbek

1 Introduction

Named Entity Recognition (NER) is a key task in natural language processing [19], and its goal is to identify entities representing names of people, places, and organizations in text. NER has wide application scenarios, such as information extraction [9], machine translation [3], question answering system [18], etc.

In recent years, NER has made great progress on high-resource languages, and many deep learning methods have achieved high accuracy [12,22,28]. However, the training of these methods relies heavily on large-scale datasets [21]. Consequently, the most significant advances in NER have been achieved in resource-rich languages such as English [20], French [25], German [4] and Chinese [8]. In

A. Yusufu and L. Jiang—Co author.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 171–183, 2023.
https://doi.org/10.1007/978-3-031-44693-1_14

contrast, in low-resource languages, the effect of NER is still poor, which limits understanding and processing of these languages to some extent. The biggest challenge in achieving high-quality NER is usually the lack of language resources, such as manually annotated datasets and pre-trained models.

The Uzbek language studied in this paper is one of the low-resource languages. The population of this language is about 30 million, most of them are located in Uzbekistan, and the rest are scattered in Central Asian countries and Xinjiang, China, but relatively little research has been done on natural language processing for this language. The difficulty of realizing Uzbek NER lies in the limited scale of academic datasets of the language, and the lack of large-scale annotated corpus. In order to solve this problem and promote the research of Uzbek NER, we constructed a large-scale human-annotated Uzbek named entity corpus. To address the issue of entity sparsity, we reviewed the corpus and only kept sentences that contained three or more entities. It contains nearly 11,366 sentences, covering three entity types: person name, place name, and organization name.

NER can be solved by various methods, such as sequence labeling [24], span enumeration [22], hypergraph [16] and sequence-to-sequence [28] and grid tagging [12]. Because the main goal of this paper is built a Uzbek NER dataset and set up a strong baseline, we select one of the state-of-the-art (SoTA) NER model based on grid tagging as our baseline. Grounded on this model, we consider the characteristics of Uzbek and extend it by incorporating unique affix feature information of the language and expanding the training corpus by translating Cyrillic text into Latin.

Moreover, BERT [6] and BiLSTM [10] are used to provide contextualized word representations, combine them with affix feature representations to form a 2D grid of word pairs, and use multi-grained 2D convolutions to reproduce word pair representations. Finally, we employ a common predictor using dual-effect and multi-layer perceptron classifiers to generate all possible entity mentions. Our results show significant performance improvements in Uzbek NER.

In comparison to four baseline models, our proposed model [1]outperforms them by improving F1 scores by 0.34%, and the grid-tagging-based method performs better due to its attention to both entity boundary and information inside. Our model improves performance by 0.46% and 0.58% when adding affix features and augmenting the corpus with translation data, respectively.

Our contributions are as follows: 1) We constructed the first high-quality Uzbek NER corpus; 2) We introduced affix features and adopted data augmentation methods to improve the performance of NER in Uzbek; 3) Our model outperformed existing methods, achieves the state-of-the-art performance, and sets a new benchmark for the Uzbek NER task.

Our work shows that for low-resource language NER tasks, data augmentation and feature engineering are also two improvement directions. Abundant data and knowledge can help the model to learn a more generalized language representation, overcome the limitations of data scarcity, and thus greatly improve the

[1] Code is available at https://github.com/azhar520/NER.

performance of the model. This provides a strong reference for further advancing low-resource language processing.

2 Related Work

In low-resource scenarios, named entity recognition faces some challenges, such as the lack of large-scale annotation data, the quality of annotation data, and the consistency of annotation standards. In order to solve these problems, the research of low-resource entity naming recognition has emerged in recent years.

In past studies, many researchers have explored the use of cross-language transfer to solve the problem of low-resource named entity recognition. These studies show that using existing high-resource language annotated data to train a model and then transferring the model to a low-resource language can effectively improve the performance of named entity recognition in low-resource languages. For example [11] and others used the method of transfer learning to perform named entity recognition on Indonesian, and the results showed that the method performed better than the baseline model on low-resource languages. Similarly, Sun et al. (2018) [23] migrated an English entity recognition model to Hungarian and Italian, and achieved good results.

In addition to cross-language transfer, some researchers have explored human-annotated low-resource named entity recognition methods. This approach trains high-quality named entity recognition models by leveraging expert annotators to annotate a small amount of data. For example, Al-Thubaity et al. (2022) [2] used human-annotated Arabic datasets to train named entity recognition models and achieved good results. Similarly, Truong et al. (2021) [26] used a manually annotated Vietnam dataset to train an named entity recognition model and achieved higher performance than the baseline on the test set. In addition to the above methods, some researchers have explored the method of combining cross-language transfer and human annotation. This method uses cross-language transfer to leverage knowledge of high-resource languages, and then uses a small amount of human-labeled data to tune the model to achieve better results. For example, Adelani et al. (2021) [1] used cross-lingual transfer and human-annotated data to solve the problem of named entity recognition in African languages and achieved higher performance than the baseline.

Uzbek belongs to the Altaic language family and has its own grammar and rich morphological structure. Therefore, there are some special problems and challenges in the field of Uzbek NER. Although there are some researches on Uzbek natural language processing, it does not specifically involve the field of named entity and recognition. In order to fill this gap, we have done three aspects of work. First, we constructed a news-based Uzbek named entity and recognition corpus. Second, we increased the number of entity placeholders for Uzbek Cyrillic to Uzbek Latin multilingual conversion of the corpus to increase the diversity of the data set, reduce the risk of over fitting, and improve cross-language performance. Thirdly, we conducted various experiments on the corpus and incorporated affix features to provide morphological-level knowledge, with the aim of enhancing the accuracy and robustness of our entity recognition system.

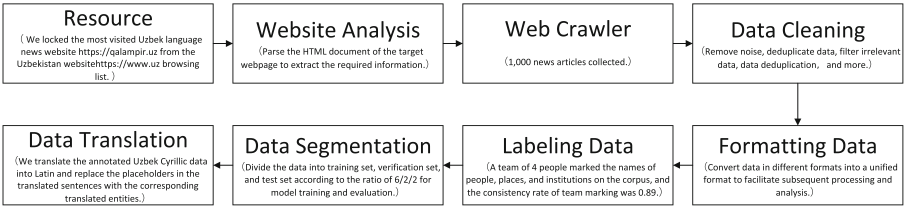

Resource
（We locked the most visited Uzbek language news website https://qalampir.uz from the Uzbekistan websitehttps://www.uz browsing list.）
Website Analysis
（Parse the HTML document of the target webpage to extract the required information.）
Web Crawler
（1,000 news articles collected.）
Data Cleaning
（Remove noise, deduplicate data, filter irrelevant data, data deduplication, and more.）
Formatting Data
（Convert data in different formats into a unified format to facilitate subsequent processing and analysis.）
Labeling Data
（A team of 4 people marked the names of people, places, and institutions on the corpus, and the consistency rate of team marking was 0.89.）
Data Segmentation
（Divide the data into training set, verification set, and test set according to the ratio of 6/2/2 for model training and evaluation.）
Data Translation
（We translate the annotated Uzbek Cyrillic data into Latin and replace the placeholders in the translated sentences with the corresponding translated entities.）

Fig. 1. Our overall workflow for dataset construction.

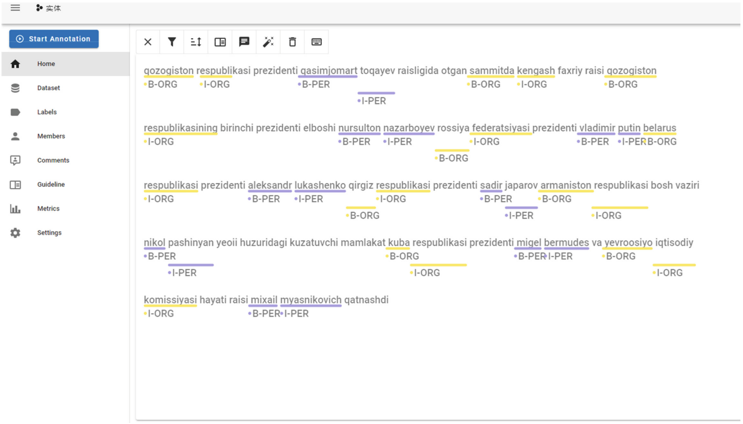
3 Dataset Construction

3.1 Data Collection and Preprocessing

First, we locked the most visited Uzbek language news website[2] from the Uzbekistan website[3] browsing list. We then analyzed the website to determine the information that we wanted to crawl, which included the title, text, time, and author of news articles. 1,000 news articles were collected by web crawler. The data was then cleaned to remove HTML tags, useless characters, and other extraneous information. This was done to ensure that the data was in a consistent format and ready for subsequent analysis. Then, the cleaned data was stored in a database or a text file to facilitate subsequent analysis and processing. Finally, we obtained the original corpus consisting of 49,019 sentences. The flowchart is shown in Fig. 1.

3.2 Data Annotation and Postprocessing

Our corpus is annotated by 4 annotators, two men and two women. They are graduate students, linguistics majors, non-native speakers but proficient in



qozogiston respublikasi prezidenti qasimjomart toqayev raisligida otgan sammitda kengash faxriy raisi qozogiston

respublikasining birinchi prezidenti elboshi nursulton nazarboyev rossiya federatsiyasi prezidenti vladimir putin belarus

respublikasi prezidenti aleksandr lukashenko qirgiz respublikasi prezidenti sadir japarov armaniston respublikasi bosh vaziri

nikol pashinyan yeoii huzuridagi kuzatuvchi mamlakat kuba respublikasi prezidenti migel bermudes va yevroosiyo iqtisodiy

komissiyasi hayati raisi mixail myasnikovich qatnashdi

Fig. 2. Annotation schema and platform.

[2] https://qalampir.uz.

[3] https://www.uz.

Uzbek. After nearly 3 months, the annotation was completed on the doccano platform. The tool supports visualization and is easy to use. A typical example is shown in Fig. 2.

Specific steps are as follows: First, we trained the annotators, informed them of the purpose of annotation, and the specific task content, and showed annotation cases for them to learn and discuss, answered their questions, and finally explained the operating specifications of the annotation system, and precautions.

Secondly, we divided the data into 4 parts, and divided the annotators into 2 groups with male and female collocations. Each person marked a piece of data. After each completed, the members of the same group exchanged data with each other for cross-labeling. Inter-annotator agreement was 0.89, as measured by span-level Cohen's Kappa [5]. We organize four people to discuss the inconsistency and until a consensus is reached.

Finally, we traversed all the data to check and corrected a few errors. The annotator training process took about a week. After annotator training, we provide each annotator with a batch for a period of one month. Then, we asked them to exchange data with a team member to start a new round of annotations for a month, unless they were tired, bored, or sick. We do this to ensure the quality of annotations. They were checked for consistency after two months, inconsistencies were dealt with together, and then another review was conducted, and thus, after nearly three months, we finally obtained a golden corpus with 49,019 labels consisting of 879,822 tokens, including 24,724 names of people, 35,743 names of places, and 25,697 names of institutions.

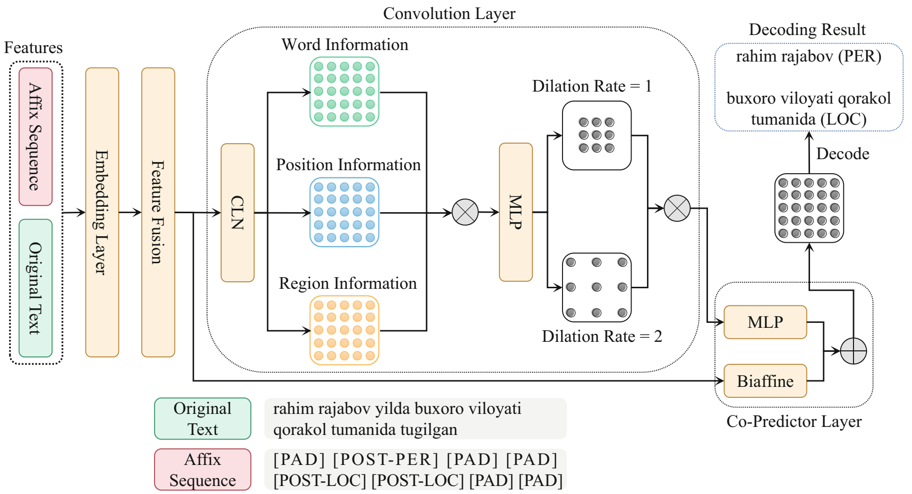


Fig. 3. Our model architecture. $\boldsymbol{H}^x$ and $\boldsymbol{H}^c$ represent the original text embedding and the affix sequence embedding respectively. $\oplus$ and $\otimes$ represent element-wise addition and concatenation operations. Both the convolutional layer and the collaborative prediction layer come from the SoTA model [12].

Due to the sparsity of entity data in news sentences, we retain sentences with three or more entities in a sentence. After screening, we got a corpus of 11,366 sentences consisting of 402,707 tokens. The longest sentence has 56 words, the shortest sentence has only 5 words, and the average length is 35.4 tokens. The corpus contains a total of 78,385 named entities. Among them, the longest name of a person is composed of 5 tokens, the name of a place is composed of 4 tokens, and the name of an organization is composed of 14 tokens. We randomly divide the gold-marked corpus into training sets/verification set/testing set, the ratio is 6/2/2, and the statistical results of the corpus are shown in the Table 1.

Table 1. Dataset statistics

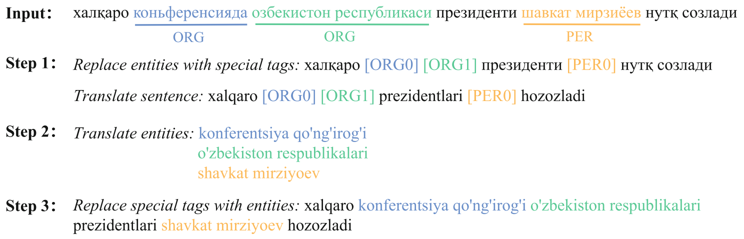
Data	**Sentence**	**Token**	**Token/Sent**	**PER**	**LOC**	**ORG**	**Discontinuous**
Train	7,366	311,885	42.35	14,137	11,029	12,084	4.26%
Dev	2,000	45,035	22.50	3,094	3,974	3,142	3.16%
Test	2,000	45,787	22.90	4,252	3,199	3,054	3.66%

4 Method

Our model is an improvement on W^{2}NER [12], including embedding layer, convolution layer and co-predictor layer. The difference is that the affix feature is integrated into the model. Our model architecture is shown in Fig. 3.

4.1 Features

Before introducing the model, we briefly introduce the construction of the affix sequence. Uzbek language contains many affix features, which are helpful for identifying entities. Therefore, we count the corresponding affixes according to the type. During the labeling process, we found that Uzbek personal names usually end with "ov", "ova", and "lar"; place names usually end with "stan", "shahr", "ko'li", etc.; institution names often end with "markaz", "kompaniya", "tashkilot" and other affix endings. These affixes include 64 place name prefixes, 23 personal name prefixes, 24 personal name suffixes, and 105 organizational name suffixes. Based on our statistics, we use four special tags to represent affix features, namely *[PRE-PER]* (PER class prefix), *[POST-PER]* (*PER* class postfix), *[POST-LOC]* (LOC class postfix) and *[POST-ORG]* (ORG class postfix). If there is no affix in the token, it will be filled with *[PAD]*. In this way, we can construct the affix sequence corresponding to the original text, such as the example shown in the bottom part of Fig. 3, an original text and its corresponding affix sequence.



Input: халқаро коньференсияда озбекистон республикаси президенти шавкат мирзиёев нутқ созлади

Step 1: *Replace entities with special tags:* халқаро [ORG0] [ORG1] президенти [PER0] нутқ созлади

Translate sentence: xalqaro [ORG0] [ORG1] prezidentlari [PER0] hozozladi

Step 2: *Translate entities:* konferentsiya qo'ng'irog'i
o'zbekiston respublikalari
shavkat mirziyoev

Step 3: *Replace special tags with entities:* xalqaro konferentsiya qo'ng'irog'i o'zbekiston respublikalari prezidentlari shavkat mirziyoev hozozladi

Fig. 4. Flowchart for data translation. Input is Cyrillic, and finally translated into Latin.

4.2 Data Translation

Since Uzbek includes Latin and Cyrillic, inspired by Liu et al. [14], in the data preprocessing stage, we consider translating Cyrillic to Latin to augment the training corpus. The specific translation process is divided into three steps: first, replace the entities in the Cyrillic sentence with special tags, and then translate into Latin; then translate the Cyrillic entities into Latin one by one; finally fill in the Latin entities into the translated Latin sentence. The whole process is translated using the Google translation model, and the overall flow chart is shown in Fig. 4.

4.3 NER Model

Embedding Layer. The embedding layer is the same as W^{2}NER, including BERT [6] and BiLSTM [10], but the input not only has the original text $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$ of length n, but also the affix sequence $C = \{c_1, c_2, \ldots, c_n\} \in \mathbb{R}^n$. After the embedding layer, the original text embedding $\boldsymbol{H}^x$ and the affix sequence embedding $\boldsymbol{H}^c$ are obtained:

$$\begin{aligned} \boldsymbol{H}^x &= \{\boldsymbol{h}_1^x, \boldsymbol{h}_2^x, \ldots, \boldsymbol{h}_n^x\} \in \mathbb{R}^{n \times d_h}, \\ \boldsymbol{H}^c &= \{\boldsymbol{h}_1^c, \boldsymbol{h}_2^c, \ldots, \boldsymbol{h}_n^c\} \in \mathbb{R}^{n \times d_h}, \end{aligned} \tag{1}$$

where $\boldsymbol{h}_i^x$, $\boldsymbol{h}_i^c \in \mathbb{R}^{d_h}$ are the representations of the i-th token, and d_h represents the dimension of a token representation.

After that, we sum $\boldsymbol{H}^x$ and $\boldsymbol{H}^c$ at the element level to get the text embedding $\boldsymbol{H}^s = \{\boldsymbol{h}_1^s, \boldsymbol{h}_2^s, \ldots, \boldsymbol{h}_n^s\} \in \mathbb{R}^{n \times d_h}$ that incorporates affix features. The subsequent process is the same as W^{2}NER, so we will only briefly introduce it.

Convolution Layer. After obtaining the text embedding $\boldsymbol{H}^s$ that incorporates the affix feature, the Conditional Layer Normalization (CLN) mechanism is used to generate a 2D grid $\boldsymbol{V}$, where each item $\boldsymbol{V}_{ij}$ in $\boldsymbol{V}$ is a representation of a word pair (x_i, x_j), so:

$$\boldsymbol{V}_{ij} = \text{CLN}(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s) = \gamma_{ij} \odot (\frac{\boldsymbol{h}_j^s - \mu}{\sigma}) + \lambda_{ij}, \tag{2}$$

where $\boldsymbol{h}_i$ is the condition to generate the gain parameter $\gamma_{ij} = \boldsymbol{W}_\alpha \boldsymbol{h}_i^s + b_\alpha$ and bias $\lambda_{ij} = \boldsymbol{W}_\beta \boldsymbol{h}_i^s + b_\beta$ of layer normalization. $\boldsymbol{W}_\alpha$, $\boldsymbol{W}_\beta \in \mathbb{R}^{d_h \times d_h}$ and $\boldsymbol{b}_\alpha$, $\boldsymbol{b}_\beta \in \mathbb{R}^{d_h}$ are trainable weights and biases respectively. μ and σ are the mean and standard deviation across the elements of $\boldsymbol{h}_j^s$.

Then word, position and sentence information on the grid is modeled, where $\boldsymbol{V} \in \mathbb{R}^{n \times n \times d_h}$ represents word information, $\boldsymbol{V}^p \in \mathbb{R}^{n \times n \times d_{h_p}}$ represents the relative position information between each pair of words, and $\boldsymbol{V}^r \in \mathbb{R}^{n \times n \times d_{h_r}}$ represents the region information for distinguishing lower and upper triangle regions in the grid. They are concatenated them to get the position-region aware representation of the grid:

$$\boldsymbol{Z} = \mathrm{MLP}_1([\boldsymbol{V}; \boldsymbol{V}^p; \boldsymbol{V}^r]) \in \mathbb{R}^{n \times n \times d_{h_z}}, \tag{3}$$

Finally, the multiple 2D dilated convolutions (DConv) with different dilation rates are used to capture the interactions between the words with different distances, formulated as:

$$\boldsymbol{Q} = \mathrm{GeLU}(\mathrm{DConv}(\boldsymbol{Z})), \tag{4}$$

where $\boldsymbol{Q} \in \mathbb{R}^{N \times N \times d_q}$ is the output and GeLU is a activation function.

Co-Predictor Module. Finally, the word pair relationship is predicted by the co-predictor, which includes the MLP predictor and the biaffine predictor. Therefore, we take these two predictors to calculate the two independent relationship distributions (x_i, x_j) of word pairs at the same time, and combine them as the final prediction. For MLP, the relationship score of each word pair (x_i, x_j) is calculated as:

$$\boldsymbol{y}_{ij}^{'} = \mathrm{MLP}_2(\boldsymbol{Q}_{ij}), \tag{5}$$

The input of the biaffine predictor is the input $\boldsymbol{H}^s$ of the CLN, which can be considered as a residual connection. Two MLPs are used to calculate the representation of each word in the word pair (x_i, x_j). Then, the relationship score between word pairs (x_i, x_j) is calculated using a biaffine classifier:

$$\boldsymbol{y}_{ij}^{''} = \boldsymbol{s}_i^\top \boldsymbol{U} \boldsymbol{o}_j + \boldsymbol{W}[\boldsymbol{s}_i; \boldsymbol{o}_j] + \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{U}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters, and $\boldsymbol{s}_i = \mathrm{MLP}_3(\boldsymbol{h}_i^s)$ and $\boldsymbol{o}_j = \mathrm{MLP}_4(\boldsymbol{h}_j^o)$ represent the subject and object representations respectively. Finally, we combine the scores from the MLP and biaffine predictors to get the final score:

$$\boldsymbol{y}_{ij} = \mathrm{Softmax}(\boldsymbol{y}_{ij}^{'} + \boldsymbol{y}_{ij}^{''}). \tag{7}$$

Decoding Algorithm. We decode entities based on two designed word pair relationships, which are (1) Next-Neighboring-Word (NNW) indicates that the word pair (x_i, x_j) belongs to an entity, and the next word of x_i in the entity is x_j. (2) Tail-Head-Word-* (THW-*) indicates that the word in the row of the grid is the tail of the entity, and the word in the column of the grid is the head of the entity. $*$ indicates the entity type.

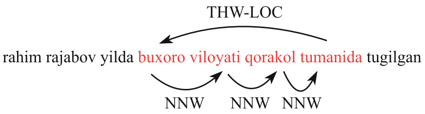

THW-LOC
rahim rajabov yilda buxoro viloyati qorakol tumanida tugilgan
NNW
NNW
NNW

Fig. 5. An example showing the process of identifying entities.

We also provided an example in Fig. 5 to explain the process of identifying different types of entities. For example, for the PER entity "oydin", it can be known from the THW-PER relationship that "oydin" is both the head and the tail of an entity, so it itself is an entity with a length of 1 and its category is PER. Then, for the ORG entity "ozbekiston respublikasi oliy majlis", by using the NNW relationship with the subject "ozbekiston" and object "respublikasi", we recognize "ozbekiston respublikasi" as a part of the entity. Similarly, "respublikasi oliy" and "oliy majlis" is also recognized in the same way. Then, by using the THW-ORG, we recognize "ozbekiston" and "majlis" are the head and tail of the entity, so that "ozbekiston respublikasi oliy majlis" can be recognized completely and its category is ORG.

5 Experiments

5.1 Experimental Setting

We conduct experiments on our UzNER (Latin) dataset to evaluate the effectiveness of our proposed model. If the token sequence and type of a predicted entity are exactly the same as those of a gold entity, the predicted entity is regarded as true-positive. We run each experiment three times and report their average value.

Our model uses *bert-base-multilingual-cased* [6] as the backbone network. We set a dropout of 0.5 on both the output representations of the BERT and convolutional module, and a dropout of 0.33 on the output representations of the co-predictor module, the learning rate of BERT and the learning rate of other modules are 1e-5 and 1e-3 respectively, the batch size is 12, and d_q can choose 64, 80, 96 and 128. The hyper-parameters are adjusted according to the fine-tuning on the development sets.

5.2 Baselines

We use some existing models of different methods as our baseline models. All baseline models are trained using an expanded corpus after translation. In addition, since our corpus is in Uzbek, the backbone network uses multilingual pre trained models.

BiLSTM+CRF [10] is the most basic sequence labeling model. Due to the presence of discontinuous entities in the dataset, we use BIOHD [24] tags to decode the entities. **BartNER** [28] is based on the Seq2Seq method, and they use pre-trained language models to solve NER tasks. We use *mbart-large-cc25* [15] as the backbone network. **W^{2}NER** [12] is based on a grid labeling method, which identifies all possible entities through word pair relationships. We use *bert-base-multilingual-cased* [6] as the backbone network. **UIE** [17] is a unified text-to-structure generation framework. UIE is not pre-trained in our experiments. We use *mt5-base* [27] as the backbone network.

5.3 Comparison with Baselines

The comparison results with the baseline models are shown in Table 2. We have the following findings: 1) Our model outperforms four baseline models. Compared with the method of Li et al. (2022) [12], our method improves the F1s by 0.34; 2) The grid-tagging-based method outperforms other methods, because the method not only pays attention to the boundary of the entity, but also pays attention to the information inside the entity. 3) The effect of BiLSTM+CRF is the worst, which is natural, because its structure is too simple compared to other models, and it can learn too little knowledge.

Table 2. Comparison with baseline models and ablation experiments. Green scores represent the best result in that column, blue scores represent the second best result in that column excluding ablation results.

	P	**R**	**F1**
BiLSTM+CRF [10]	86.81	79.28	82.87
w/o Data Translation	84.62	80.05	82.27
BartNER [28]	92.34	90.16	91.23
w/o Data Translation	90.42	88.09	89.24
W^{2}NER [12]	92.47	90.29	91.37
w/o Data Translation	92.35	90.01	91.16
UIE [17]	91.28	87.13	89.16
w/o Data Translation	87.23	83.41	85.28
Ours	**92.83**	**90.63**	**91.71**
w/o Affix	92.39	90.13	91.25
w/o Data Translation	91.93	90.34	91.13

5.4 Ablation Studies

The results of the ablation experiments are shown in Table 2. We mainly analyzed the two improvement schemes that we proposed. First, when we remove the affix features, the performance of our model on F1s drops by 0.46, indicating that the fusion of affix features is effective and can better improve the performance of the model. We then train the model without translation data, and our model performance drops by 0.58 on F1s, which is natural, more data allows the model to learn more features. In addition, for the baseline model, we also train on the augmented data without translation, and the performance is also reduced.

Table 3. Performance comparison among different entity classes. Green scores and blue scores represent the best and second best results in that column excluding ablation results.

	LOC			PER			ORG		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BiLSTM+CRF [10]	90.77	87.42	89.06	86.30	73.73	79.52	82.55	78.80	80.63
w/o Data Translation	89.03	87.09	88.05	85.70	76.46	80.82	78.39	78.28	78.33
BartNER [28]	94.42	**95.79**	95.10	**91.32**	85.37	**88.24**	89.80	91.88	90.83
w/o Data Translation	93.22	95.08	94.14	88.84	82.77	85.70	87.82	89.54	88.67
W^{2}NER [12]	95.85	94.05	94.94	90.02	85.74	87.83	91.80	92.23	92.01
w/o Data Translation	95.73	93.92	94.81	90.28	85.60	87.87	91.64	92.09	91.86
UIE [17]	94.85	90.62	92.69	91.06	81.70	86.13	91.06	89.08	90.06
w/o Data Translation	91.47	87.06	89.21	86.96	75.95	81.08	86.80	85.79	86.29
Ours	**96.41**	94.42	**95.41**	90.08	**86.45**	88.22	**92.83**	**92.52**	**92.67**
w/o Affix	95.92	94.28	95.09	89.77	85.47	87.57	92.26	92.33	92.29
w/o Data Translation	95.39	94.06	94.72	89.63	86.33	87.95	91.45	92.07	91.76

5.5 Performance Analysis on Different Entity Types

We also explored the effectiveness of our method and baseline method on three entity classes. The comparison results with the baseline model are shown in Table 3. First, our method outperforms all baseline models on LOC and ORG by 0.31 and 0.66 on F1s compared to the second-best results. The performance achieved on F1s is the second best compared to the baseline model, only 0.02 lower than the best performance.

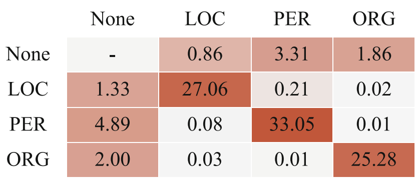
Table 4. Error analysis experiment. EBE and ETE represent Entity Boundary Error and Entity Type Error, respectively.

Error Type	EBE	ETE
All (%)	99.65	0.35
LOC (%)	22.29	0.11
PER (%)	47.48	0.21
ORG (%)	29.87	0.03

In the lower part of Table 3, we also analyze the ablation results on different entity classes. First, with the removal of affix features, the performance of our model on all three types of entities degrades. Then, without training the model with translated data, the performance of our model drops on all three types of entities. Finally, the performance of the baseline model on all three types of entities also decreases when trained without translation data.

5.6 Error Analysis

We also performed error analysis to learn more about our model. The results are shown in the of Table 4. Most of the errors come from boundary errors, accounting for 99.65% of all errors, because entity boundaries are difficult to identify, which is a well-known problem in previous work [7,13]. In addition, we also analyzed the proportion of different types of errors. Regardless of the type of error, the PER entity has the largest proportion of errors. This is because PER has higher text diversity and the model is more difficult to predict more PER entities. Finally, Fig. 6 is a heat map of the confusion matrix of error analysis. The diagonal line represents the proportion of correct recognition, so it is the highest proportion, which is natural. In addition, the proportion of the first row and the first column is next, which is reasonable, because the proportion of these two parts is equivalent to the boundary error, which is consistent with the results in Table 4.



	None	LOC	PER	ORG
None	-	0.86	3.31	1.86
LOC	1.33	27.06	0.21	0.02
PER	4.89	0.08	33.05	0.01
ORG	2.00	0.03	0.01	25.28

Fig. 6. The confusion matrix for error analysis. None represents non-entity. Numbers represent percentages. Rows and columns represent the gold and predicted results, respectively.

6 Conclusion

Our study proposes a novel approach to enhance the state-of-the-art model for Uzbek NER by incorporating unique affix feature information of the language and expanding the training corpus by translating Cyrillic text into Latin. Our proposed model outperforms four baseline models with a significant F1 score improvement of 0.34%, demonstrating the effectiveness of our approach. The grid-tagging-based method is found to be superior to other methods due to its attention to both entity boundary and information inside. Our findings highlight the importance of incorporating unique language features and utilizing advanced neural network architectures for NER tasks. In the future, further exploration of other language-specific features and integration of cross-lingual transfer learning can potentially improve the performance of NER models for low-resource languages like Uzbek.

Acknowledgment. This work is supported by the Natural Science Program of Xinjiang Uygur Autonomous Region for the Construction of Innovation Environment (Talents and Bases) (Special Training of Scientific and Technological Talents of Ethnic Minorities)(2022D03001), the National Natural Science Foundation of China (No. 62176187; No. 61662081), the Major Projects of the National Social Science Foundation of China (No.11&ZD189; No.14AZD11), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No. 18JZD015).

References

1. Adelani, D.I., et al.: Masakhaner: named entity recognition for African languages. Trans. Assoc. Comput. Linguist. **9**, 1116–1131 (2021)
2. Al-Thubaity, A., Alkhereyf, S., Alzahrani, W., Bahanshal, A.: Caraner: the Covid-19 Arabic named entity corpus. In: WANLP@EMNLP 2022, pp. 1–10 (2022)
3. Balabantaray, R.: Name entity recognition in machine translation. Emerg. Technol **1**(3), 3 (2010)
4. Benikova, D., Biemann, C., Reznicek, M.: Nosta-d named entity annotation for German: Guidelines and dataset. In: LREC 2014, pp. 2524–2531 (2014)
5. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**(1), 37–46 (1960)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT 2019, pp. 4171–4186 (2019)
7. Fei, H., Ji, D., Li, B., Liu, Y., Ren, Y., Li, F.: Rethinking boundaries: end-to-end recognition of discontinuous mentions with pointer networks. In: AAAI 2021, pp. 12785–12793 (2021)
8. Ji, B., et al.: A hybrid approach for named entity recognition in Chinese electronic medical record. BMC Med. Inform. Decis. Mak. **19**(2), 149–158 (2019)
9. Krallinger, M., Valencia, A.: Text-mining and information-retrieval services for molecular biology. Genome Biol. **6**(7), 1–8 (2005)
10. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: HLT-NAACL 2016, pp. 260–270 (2016)
11. Leonandya, R., Ikhwantri, F.: Pretrained language model transfer on neural named entity recognition in indonesian conversational texts. arXiv preprint arXiv:1902.07938 (2019)

12. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: AAAI 2022. vol. 36, pp. 10965–10973 (2022)
13. Liu, J., et al.: TOE: a grid-tagging discontinuous NER model enhanced by embedding tag/word relations and more fine-grained tags. IEEE/ACM Trans. Audio, Speech, Lang. Process. **31**, 177–187 (2022)
14. Liu, L., Ding, B., Bing, L., Joty, S., Si, L., Miao, C.: Mulda: a multilingual data augmentation framework for low-resource cross-lingual NER. In: ACL/IJCNLP 2021, pp. 5834–5846 (2021)
15. Liu, Y., et al.: Multilingual denoising pre-training for neural machine translation. Trans. Assoc. Comput. Linguist. **8**, 726–742 (2020)
16. Lu, W., Roth, D.: Joint mention extraction and classification with mention hypergraphs. In: EMNLP 2015, pp. 857–867 (2015)
17. Lu, Y., et al.: Unified structure generation for universal information extraction. In: ACL 2022, pp. 5755–5772 (2022)
18. Mollá, D., Van Zaanen, M., Smith, D.: Named entity recognition for question answering. In: ALTA 2006, pp. 51–58 (2006)
19. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
20. Ringland, N., Dai, X., Hachey, B., Karimi, S., Paris, C., Curran, J.R.: NNE: a dataset for nested named entity recognition in english newswire. arXiv preprint arXiv:1906.01359 (2019)
21. Rosenfeld, J.S.: Scaling laws for deep learning. arXiv preprint arXiv:2108.07686 (2021)
22. Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., Lu, W.: Locate and label: a two-stage identifier for nested named entity recognition. In: ACL/IJCNLP 2021, pp. 2782–2794 (2021)
23. Sun, P., Yang, X., Zhao, X., Wang, Z.: An overview of named entity recognition. In: IALP 2018, pp. 273–278. IEEE (2018)
24. Tang, B., Hu, J., Wang, X., Chen, Q.: Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF. In: Proceedings of the Wireless Communications and Mobile Computing 2018 (2018)
25. Tedeschi, S., Maiorca, V., Campolungo, N., Cecconi, F., Navigli, R.: Wikineural: combined neural and knowledge-based silver data creation for multilingual NER. In: EMNLP (Findings) 2021, pp. 2521–2533 (2021)
26. Truong, T.H., Dao, M.H., Nguyen, D.Q.: Covid-19 named entity recognition for Vietnamese. arXiv preprint arXiv:2104.03879 (2021)
27. Xue, L., et al.: mt5: A massively multilingual pre-trained text-to-text transformer. In: NAACL-HLT 2021, pp. 483–498 (2021)
28. Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., Qiu, X.: A unified generative framework for various NER subtasks. In: ACL/IJCNLP 2021, pp. 5808–5822 (2021)