



Article Learning Subword Embedding to Improve Uyghur Named-Entity Recognition

Alimu Saimaiti^{1,2,3}, Lulu Wang^{1,2} and Tuergen Yibulayin^{1,2,*}

- ¹ College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; 107556516097@stu.xju.edu.cn (A.S.); wanglulu@stu.xju.edu.cn (L.W.)
- ² Multilingual Information Technology Laboratory of Xinjiang University, Urumqi 830046, China
- ³ Iflytek Voice and Language Joint Laboratory, Xinjiang University, Urumqi 830046, China
- * Correspondence: turgun@xju.edu.cn; Tel.: +86-138-9981-0133

Received: 27 March 2019; Accepted: 11 April 2019; Published: 15 April 2019



Abstract: Uyghur is a morphologically rich and typical agglutinating language, and morphological segmentation affects the performance of Uyghur named-entity recognition (NER). Common Uyghur NER systems use the word sequence as input and rely heavily on feature engineering. However, semantic information cannot be fully learned and will easily suffer from data sparsity arising from morphological processes when only the word sequence is considered. To solve this problem, we provide a neural network architecture employing subword embedding with character embedding based on a bidirectional long short-term memory network with a conditional random field layer. Our experiments show that subword embedding can effectively enhance the performance of the Uyghur NER, and the proposed method outperforms the model-based word sequence method.

Keywords: subword embedding; Uyghur; named-entity recognition; morphological processing; word sequence; natural language processing; deep learning; word-based neural model

1. Introduction

Many scholars study named-entity recognition (NER) because of its importance to natural language processing. NER uses sequence-labeling to automatically recognize entities in text, including persons, locations, and organizations. Using deep learning, NER has achieved good performance with languages having large-scale datasets, such as English [1,2] and Chinese [3,4]. Strengthening the information construction of ethnic minority languages is a driving force in the development and social advancement of China. However, because Uyghur is an ethnic minority language in China, NER as a fundamental information construction task requires vast improvements. The main problem is that Uyghur is a morphologically rich and typical agglutinating language, wherein a word may present different variations with the connection of affixes. Thus, the complex and rich morphology presents the problem of extremely sparse data. Moreover, the beginning characters of named entities have no capitalization that can be used as distinct features, unlike that English. Additionally, there are only a small quantity of annotated corpora and no public corpus for Uyghur NER.

Currently, most research on Uyghur NER has adopted statistical methods of machine learning, including conditional random fields (CRF) [5] and hybrid approaches [6,7]. These methods depend excessively on handcrafted features and domain-specific knowledge resources. However, the process of collecting features and resources is inefficient and expensive. To avoid heavy feature engineering, our objective is to provide a neural network architecture that employs subword- and character-embedding based on a bidirectional long short-term memory (LSTM) network with a CRF layer to improve Uyghur NER performance.

2. Related Works

NER for English and German on the CoNLL-2003 dataset has drawn the attention of many researchers. Traditionally, NER systems have employed machine-learning tactics, including CRF [8], hidden Markov models [9] and support vector machines [10]. Handcrafted features and domain-specific knowledge resources (e.g., a manually annotated dataset) are needed as inputs to train these models.

With advances in deep learning, neural network models for sequence labeling have been spectacularly well-utilized for high-performance NER tasks. Collobert et al. [11] adopted an architecture based on convolutional neural networks (CNNs) with CRFs to solve sequence-tagging problems, which improved the performance and significantly reduced the dependency on task-specific engineering. Huang et al. [12] proposed a bidirectional (bi) LSTM with a CRF Layer, achieving 90.10% F1 with both Senna embedding and gazetteer features. A bi-LSTM-CNN architecture was used to detect word- and character-level features, as proposed by Chiu et al. [13]. It outperformed the method that relied on the heavy feature engineering and achieved fairly good performance on CoNLL-2003 and OntoNotes 5.0. Lample et al. [1] presented a bi-LSTM-CRF architecture that obtained effective information from character-based word embedding. Rei et al. [14] presented an architecture that amalgamates character-based word embedding by using an attention mechanism, surpassing the architecture based on concatenating the word- and character-level representations. Ma et al. [2] offered a bi-LSTM-CNN-CRF neural network architecture that automatically benefits from word- and character-level representation. Shen et al. [15] used deep active learning for NER, but only on small-scale labeled data. However, these approaches are not particularly applicable to the morphologically rich and agglutinating languages, so many scholars have made certain improvements to take to a more ideal performance, which employed morphological embedding [16], phonological character representations [17] and morpheme-level representations [18], respectively.

Research on Uyghur NER is still in its early stages and mainly focuses on particular entities. For example, Tashpolat et al. [6] employed a CRF- and rules-based post-processing approach to achieve high performance on Uyghur person-name recognition via the analysis of agglutinative characteristics. Maimaiti et al. [7] presented a CRF model with rules for Uyghur location-name recognition by introducing different handcrafted features, especially syllables and similar words via word embedding. Maihefureti et al. [19] researched rule-based Uyghur organization-name recognition, which depended upon syntactical and semantical knowledge. Halike et al. [20] implemented the recognition of times, numerals, and quantifiers using an approach that relied on the manual rule library. Our approach is different because we simultaneously identify person, location, and organization. Recent advances notwithstanding, a morphologically rich language such as Uyghur requires a combination of word- and character-level embedding as input features, instead of handcrafted features and domain-specific knowledge.

3. Methodology

In this section, we describe the proposed neural network architecture. The word-based neural model is introduced first; it is a bi-LSTM-CRF model that promotes the performance of Uyghur NER. Then, we propose the subword-based neural model, which takes a sequence of subwords as input. To fully understand the architectures, we take a sentence using Uyghur Latin script as example, "niGmEt beyjiNdiki turalGusida turwatidu", which means "niGmEt lives in Beijing".

3.1. Word-Based Neural Model

We first introduce the word-based neural model, following the models presented by Lample et al. [1]. Figure 1 shows the neural network architecture.

Recurrent neural networks (RNNs) are neural network language models used for processing sequential data. RNNs can capture long-distance dependencies by leveraging historical information. However, they are not very effective for NER, and this causes problems of gradient vanishing and

exploding [21]. LSTMs [22] have been proposed to overcome RNN shortcomings by incorporating a memory-cell while exploiting long-term dependencies. An LSTM cell uses several gates to regulate the proportion of information to be stored vs. forgotten. Greff et al. [23] explored eight LSTM variants based on Vanilla LSTM [24] on three representative tasks and compare their performances, concluding that Vanilla LSTM performs well in all applications, while the other eight variants had no significant performance improvement. Therefore, we use the following equations to update the LSTM cell at time t, which is the same as Vanilla LSTM:

$$\mathbf{i}_{t} = \sigma(\mathbf{W}_{xi}\mathbf{x}_{t} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_{i}), \tag{1}$$

$$\check{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \qquad (2)$$

$$\boldsymbol{c}_t = (1 - \boldsymbol{i}_t) \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \check{\boldsymbol{c}}_t, \tag{3}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{xo}\boldsymbol{x}_t + \boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{co}\boldsymbol{c}_t + \boldsymbol{b}_o), \tag{4}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t), \tag{5}$$

where σ is the logistic sigmoid function; \odot indicates the point-wise product; x_t , o_t , c_t are the input, output, and cell vectors, respectively; h_t is the hidden vector at time t; W indicates the weight matrices of different gates; and b represents bias vectors. Therefore, h_t is defined by the input vector, x_t , and the hidden vector, h_{t-1} , at the previous moment.

For many sequence-labeling tasks (such as NER), both past and future information are beneficial for predictions. It is advisable to utilize bi-LSTM to capture contextual information from two directions. This method has been proven successful for many tasks [25].

For a sequence of vectors, $\mathbf{X} = (x_1, x_2, \dots, x_n)$, the bi-LSTM computes forward representations, $\vec{h} = (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n)$, and backward representations, $\vec{h} = (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n)$. Using the model, the final expression of each word is acquired by using forward and backward representations, $h_t = (\vec{h}_t, \vec{h}_t)$.

To get better feature combinations, the bi-LSTM contains a hidden layer at the top, so that we can encode a more reliable pattern for each word:

$$\boldsymbol{d}_t = \tanh(\boldsymbol{W}_d \boldsymbol{h}_t),\tag{6}$$

where W_d is a weight matrix for the hidden layer.

In general, there are two ways to estimate current labels. The first uses a softmax layer that acts as an output layer to independently make tagging decisions. The softmax function is a normalized exponential function that predicts the probability distribution over all labels with possibilities for every word:

$$p(\mathbf{y}_{t} = j | \boldsymbol{d}_{t}) = \frac{e^{W_{o,j} \boldsymbol{d}_{t}}}{\sum_{l=1}^{k} e^{e^{W_{o,l} \boldsymbol{d}_{t}}}},$$
(7)

where $p(\mathbf{y}_t = j | d_t)$ is the probability that the label of the *t*th word, \mathbf{y}_t , is *j*; *k* is the number of all possible labels; and $W_{o,j}$ is the *j*th row of the output weight matrix, W_o . During model training, the negative log-probability of the correct labeling sequence is minimized:

$$E = -\sum_{t=1}^{n} \log(\mathbf{p}(\mathbf{y}_t = j | \boldsymbol{d}_t)).$$
(8)

NER tags with "beginning-inside-outside" formats have strong constraints, meaning that an organization (ORG) on the inside cannot follow a location (LOC) at the beginning or on the outside. Thus, the softmax layer is insufficient. CRF focuses on the sentence level instead of decoding each

label independently. Thus, CRF tagging is ideal for NER tasks. Given a sequence of predictions, $\mathbf{y} = (y_1, y_2, \dots, y_n)$, its score can be defined as

$$S(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} T_{\mathbf{y}_{i}, \mathbf{y}_{i-1}} + \sum_{i=1}^{n} P_{i, \mathbf{y}_{i'}}$$
(9)

$$P_{i,y_i} = W_{o,y_i} d_i, \tag{10}$$

where *P* is the matrix of the scores output from the bi-LSTM and P_{i,y_i} is the score in which the tag of the *j*th word is y_i . $T_{y_i,y_{i-1}}$ represents the score of a transition from tag y_t to tag y_{t+1} in a sentence. Over the course of training, the log-probabilities of the correct tag-sequence are maximized:

$$\log(p(\mathbf{y}|\mathbf{X})) = \log\left(\frac{\exp(S(x,y))}{\sum_{\mathbf{y}\in\mathbf{Y}_{\mathbf{x}}}\exp(S(x,y))}\right) = S(\mathbf{X},\mathbf{y}) - \log\left(\sum_{\mathbf{y}\in\mathbf{K}}S(\mathbf{X},\mathbf{y})\right).$$
(11)

 Y_x represents the entirety of the possible tag sequences. In the test stage, we used Viterbi algorithm for prediction of the output sequence with maximal conditional probability.



Figure 1. Word-based neural model.

3.2. Subword-Based Neural Model

The input vector in the traditional bi-LSTM–CRF model takes a word as its basic unit. However, Uyghur is an agglutinating language in which a word comprises a stem and affixes. If only the word vector is considered, the semantic information cannot be fully learned, causing it to suffer from data sparsity arising from morphological processes. Therefore, we consider morphological segmentation to exploit smaller meaning-bearing units to improve performance. Morphological segmentation breaks words into meaning-bearing subword units called morphemes [26]. Thus, Uyghur morphology segmentation allows us to break words into more familiar units than have been previously observed. Uyghur morphology segmentation falls into two segmentation categories: single-point and multi-point. Single-point segmentation refers to segmenting a word into a stem and a suffix, whereas multi-point

 $\langle 0 \rangle$

segmentation refers to a more fine-grained segmentation, further segmenting a suffix on the basis of single-point segmentation. To fully explain the phenomenon, we provide the following example.

Latin Uyghur: niGmEt beyjiNdiki turalGusida turwatidu. (niGmEt lives in Beijing.) Single-point segmentation: niGmEt beyjiN/diki turalGu/sida tur/watidu Multi-point segmentation: niGmEt beyjiN/diki turalGu/si/da tur/watidu

In this study, we use three methods derived from the Xinjiang University & Iflytek Voice and Language Joint Laboratory for Uyghur morphology segmentation. The differences among the methods are shown in Table 1.

Method	Segmentation Category	F1		
bi-LSTM	single-point	90.61		
SRILM-Ngram	multi-point	43.40		
MaxMatch	multi-point	82		

Table 1. Different morphology segmentation methods for Uyghur.

To mitigate the data sparsity problem, we propose a bi-LSTM–CRF model based on the subword sequence. This model comprises bi-LSTM and CRF layers, but it is distinct from the traditional model, because its input sequence is changed, and a tag for each subword is independently predicted. Additionally, we introduce subword embedding with character embedding as the input vectors of this model. Figure 2 shows the model structure.



Figure 2. Bi-LSTM–CRF model based subword sequence with single-point segmentation.

3.3. Features

3.3.1. Word Embedding

Word embedding (i.e., distributed word representation) has become popular with researchers because of its ability to simultaneously obtain semantic and syntactic information from words in a large unlabeled corpus [27]. To obtain high-quality word embedding, instead of randomly initializing the embedding, we use a large-scale unannotated dataset to prepare pre-trained word embedding, as developed at the Xinjiang University and Iflytek Voice and Language Joint Laboratory. It contains

1,891,895 sentences and a vocabulary size of 2,461,449 tokens. We adopt the skip-gram model of word2vec, provided by Gensim (https://radimrehurek.com/gensim/index.html), while training word embedding as "pre-trained."

3.3.2. Subword Embedding

We used the above Uyghur morphology segmentation method to process the annotated dataset and took subwords as basic training units using a skip-gram model of word2vec, which is similar to training for word embedding. Thus, semantic information containing subword embedding assumes that every subword can stand independently. After segmentation, the subword vocabulary size corresponding to the bi-LSTM, SRILM-Ngram, and MaxMatch methods are 2,034,757; 2,109,530; and 2,051,620, respectively.

3.3.3. Character Embedding

Additionally, abundant structure information of the entity is embodied in character-level features. Character embedding is not only useful for researching languages rich in morphology, it also alleviates the out-of-vocabulary problem [26]. First, we randomly initialize a character lookup table with a character embedding for every character. Character-embedding matching for every character in a word is provided in both directions of the bi-LSTM network. Finally, the concatenation of the forward and backward representations from the bi-LSTM is used as the character-level feature of the word.

4. Experiments

4.1. Datasets

Our models were evaluated with a manually annotated Uyghur NER corpus, created at Multilingual Information Technology Laboratory of Xinjiang University [28]. It contains 39,027 sentences and 102,360 named entities. Person (PER), location (LOC), and organization (ORG) account for approximately 27.81%, 41.60%, and 30.58%, respectively. Additionally, the entity labels are annotated using IOB notation. We used the 10-fold cross-validation method to validate performance, where the training (train), development (dev), and test (test) sets accounted for 80%, 10%, and 10%, respectively. The statistics of the dataset are shown in Table 2.

Туре	Sentence	Token	NE	PER	LOC	ORG
dataset	39,027	1,152,645 (91,599)	102,360 (48,792)	28,469 (15,174)	42,585 (14,842)	31,306 (18,805)
train	29,270	861,967 (77,665)	76,787 (38,561)	21,304 (12,061)	32,011 (11,847)	23,472 (14,652)
dev	3902	115,689 (22,574)	10,215 (6854)	2842 (2142)	4258 (2257)	3115 (2457)
test	5855	174,989 (29,639)	15,358 (9713)	4323 (3073)	6316 (3166)	4719 (3477)

Table 2. Statistics of the entity type for the Uyghur named-entity recognition (NER) dataset.

Note: The number in parentheses indicates the number of non-repeating token or entities. Sentence, Token and NE refer to the number of sentence, tokens, named-entities in each data set.

4.2. Training and Evaluation

Our models were trained using a back-propagation algorithm that updated the parameters for every training example [1]. During the training phase, we prepared 300-dimensional pre-trained word or subword embeddings using the skip-gram model to initialize the model. We set the maximum epoch to 100. The dimensions of the forward and backward LSTMs were set to 100. We used stochastic gradient descent with a learning rate of 0.01 and a gradient clipping of 5.0 for optimization. We used dropout with a probability of 0.5 to avoid overfitting. The final dimension of our character-based embedding of words was 50. The measurement score of the Uyghur NER performance is F_1 , which relates to precision and recall on the test set.

4.3. Experimental Results and Discussion

Results for the different morphological segmentations of the subword-based neural model that only considers subword embedding are shown in Table 3. The best performance (89.02% in F-score) appeared when the Uyghur morphological segmentation-based bi-LSTM was used. However, the F_1 score of the other segment function did not show a significant improvement. The reason may be that the SRILM-Ngram- and MaxMatch-based morphological segmentation methods are a type of multi-point segmentation, causing excessive segmentation that leads to ambiguity for Uyghur NER. Furthermore, the accuracy of these two segmentation methods was relatively low. Therefore, morphological segmentation-based bi-LSTM was utilized in the next experiment.

Segmentation Method		D	ev		Test				
	PER	LOC	ORG	AVE	PER	LOC	ORG	AVE	
bi-LSTM	93.70	88.75	87.43	89.72	93.46	87.66	86.79	89.02	
SRILM-Ngram	93.42	88.73	87.46	89.65	92.72	87.16	86.16	88.42	
MaxMatch	93.14	88.49	86.93	89.29	93.11	87.26	86.85	88.78	

Table 3. Comparison of morphological segmentation on subword-based neural models (%).

Note: "Total" refer to the average F1 score for different method.

We conducted many experiments representing different models to understand their influences on the Uyghur NER system. We explored the impact of using word/subword embedding and character-level embedding. The baseline results are from Wang et al. [29], who used a semi-supervised approach based on CRF. Table 4 compares the word-based and subword-based neural models. Compared to the baseline, the neural network model has a slight advantage. We found that, when the input embedding process reached word or subword embedding, the F-score of the subword-based method was higher. When character-level embedding was added, the neural network model improved by at least 0.5% on the basis of word vectors. The word-based neural model with character-level embedding performed best for ORG. However, the results of average F1 scores show that the subword-based model was more suitable than the word-based models.

Table 4. Comparison of performance on different neural models (%). Bold indicates the best result in below models for each entity category.

Model	Input Embedding		DI	ΞV		TEST			
Wouch	1	PER	LOC	ORG	Total	PER	LOC	EST ORG 85.91 86.79 86.79 87.04 86. 79 86.79 86.79	Total
CRF (Wang et al. 2018)	-	-	-	-	-	91.65	85.72	85.91	87.43
Word-based neural model	word embedding	93.03	87.40	87.22	88.89	92.01	86.17	86.79	88.04
	+char embedding	94.47	89.19	87.82	90.24	94.63	87.80	87.04	89.49
Subword-based neural model	subword embedding	93.70	88.75	87.43	89.72	93.46	87.66	86. 79	89.02
	+char embedding	95.00	89.83	87.59	90.57	94.17	88.45	86.79	89.55

4.4. OOV Error Comparison with Different Models

To further understand the behavior of the subword-based neural model, we performed error analysis on the testing set. Specifically, we divided each dataset into in-vocabulary (IV) entities, out-of-training-vocabulary (OOTV) entities, out-of-embedding-vocabulary (OOEV) entities, and out-of-both-vocabulary (OOBV) entities. An entity is considered OOBV if at least one word is not in the training set and at least one word is in embedding vocabulary. The other three subsets can be performed the same way. Table 5 shows the statistics of the division of each corpus.

Datasets	Туре	IV	OOTV	OOEV	OOBV
Word-Datasets	DEV	10,581	4465	1607	1195
	TEST	15,718	6748	2569	1878
Subword-Datasets	DEV	13,818	4400	1844	1313
	TEST	20,497	6866	3045	2122

Table 5. Statistics of the division on each corpus.

Table 6 illustrates the performance of the subword-based and word-based neural models on diverse subsets of entities. When comparing the performance of the CRF statistical model and bi-the LSTM-CRF neural network model for each entity category, the version with only word/subword embedding had a few difficulties correctly recognizing the OOEV of named entities. This demonstrates that the neural network model largely depended on input embedding. However, the subword-based neural model with character embedding achieved a 2% improvement over the previous best OOBV result. Thus, almost all improvements of the subword-based neural model via embedding was conducive to Uyghur NER.

Table 6. Comparison of performance on different subsets of entities (%).

Model	Input Embedding –	DEV				TEST			
		IV	OOTV	OOEV	OOBV	IV	OOTV	OOEV	OOBV
Baseline	-	-	-	-	-	88.64	69.48	82.01	80.32
Word-based neural model	word embedding	97.33	79.63	79.04	77.42	96.82	77.37	74.47	75.35
	+char embedding	97.46	85.77	83.65	84.87	97.37	83.52	78.88	79.83
Subword-based neural model	subword embedding	97.54	82.26	81.33	79.87	97.32	81.13	75.90	76.79
	+char embedding	97.59	85.79	84.80	83.24	97.48	85.14	81.19	82.42

5. Conclusions

In this paper, we presented a subword-based neural network model based on bi-LSTM–CRF for Uyghur NER, which does not require handcrafted features or any knowledge sources to capture linguistic information. In experiments conducted, we utilized different Uyghur morphology segmentations and obtained very promising results compared to the word-based neural model. Further, subword embedding was conducive to system performance when the accuracy of morphology segmentation was higher, or no excessive morphology segmentation existed. Even though Uyghur is a morphologically rich and low-resource language, subword embedding is a simple and effective remedy to achieve state-of-the-art performance for such NER datasets. Further work should be done to evaluate subword embedding across other natural language processing applications, such as machine translation. Additionally, a better generic neural network model using cross-lingual embedding will be explored to deal with low-resource and agglutinating language processing.

Author Contributions: Conceptualization, A.S.; methodology, A.S.; validation, A.S. and L.W.; formal analysis, A.S.; investigation, A.S.; resources, T.Y.; data curation, A.S. and L.W.; writing—original draft preparation, A.S. and L.W.; writing—review and editing, T.Y.; supervision, T.Y.

Funding: This research was funded by the Opening Foundation of the Key Laboratory of Xinjiang Uyghur Autonomous Region of China (grant number 2018D04019); the National Natural Science Foundation of China (grant numbers 61762084, 61662077, 61462083); and the Scientific Research Program of the State Language Commission of China (grant number ZDI135-54).

Acknowledgments: The authors gratefully acknowledge all anonymous reviewers and editors for their constructive suggestions for the improvement of this paper. The authors also gratefully acknowledge fund support from Kahaerjiang Abiderexiti and data support from Maihemuti Maimaiti.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360.
- 2. Ma, X.; Hovy, E. End-to-end sequence labeling via bi-directional LSTM-CNN-CRF. *arXiv* 2016, arXiv:1603.01354.
- Dong, C.; Zhang, J.; Zong, C.; Hattori, M.; Di, H. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*; Springer: Cham, Switzerland, 2016; pp. 239–250.
- Xiang, Y. Chinese Named Entity Recognition with Character-Word Mixed Embedding. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 2055–2058.
- 5. Rozi, A.; Zong, C.; Mamateli, G.; Mahmut, R.; Hamdulla, A. Approach to recognizing Uyhgur names based on conditional random fields. *J. Tsinghua Univ.* **2013**, *53*, 873–877.
- 6. Tashpolat, N.; Wang, K.; Askar, H.; Palidan, T. Combination of statistical and rule-based approaches for Uyghur person-name recognition. *Acta Autom. Sin.* **2017**, *43*, 653–664.
- 7. Maimaiti, M.; Abiderexiti, K.; Wumaier, A.; Yibulayin, T.; Wang, L. Uyghur location names recognition based on conditional random fields and rules. *J. Chin. Inf. Process.* **2017**, *31*, 110–118.
- 8. Marcińczuk, M. Automatic construction of complex features in conditional random fields for named entities recognition. In Proceedings of the International Conference Recent Advances in Natural Language Processing, Hissar, Bulgaria, 7–9 September 2015.
- 9. Gayen, V.; Sarkar, K. An HMM based named entity recognition system for Indian languages: JU system at ICON 2013. *arXiv* **2014**, arXiv:1405.7397.
- 10. Kravalová, J.; Žabokrtský, Z. Czech Named Entity Corpus and SVM-Based Recognizer. In Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, Singapore, 7 August 2009; pp. 194–201.
- 11. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
- 12. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
- 13. Chiu, J.P.C.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist* 2016, 4, 357–370. [CrossRef]
- 14. Rei, M.; Crichton, G.K.; Pyysalo, S. Attending to characters in neural sequence labeling models. *arXiv* **2016**, arXiv:1611.04361.
- 15. Shen, Y.; Yun, H.; Lipton, Z.C.; Kronrod, Y.; Anandkumar, A. Deep active learning for named entity recognition. *arXiv* **2017**, arXiv:1707.05928.
- 16. Gungor, O.; Yildiz, E.; Uskudarli, S.; Gungor, T. Morphological embeddings for named entity recognition in morphologically rich languages. *arXiv* **2017**, arXiv:1706.00506.
- 17. Bharadwaj, A.; Mortensen, D.; Dyer, C.; Carbonell, J. Phonologically aware neural model for named entity recognition in low resource transfer settings. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1462–1472.
- Wang, W.; Bao, F.; Gao, G. Mongolian named entity recognition with bidirectional recurrent neural networks. In Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), San Jose, CA, USA, 6–8 November 2016; pp. 495–500.
- 19. Maihefureti Rouzi, M.; Aili, M.; Yibulayin, T. Uyghur organization name recognition based on syntactic and semantic knowledge. *Comput. Eng. Des.* **2014**, *35*, 2944–2948.
- 20. Halike, A.; Wumaier, H.; Yibulayin, T.; Abiderexiti, K.; Maimaiti, M. Research on recognition and translation of Chinese-Uyghur time and numeral and quantifier. *J. Chin. Inf. Process.* **2016**, *30*, 190–200.
- 21. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
- 22. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
- 23. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *28*, 2222–2232. [CrossRef] [PubMed]

- 24. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
- 25. Graves, A.; Mohamed, A.R.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
- Creutz, M.; Hirsimäki, T.; Kurimo, M.; Puurula, A.; Pylkkönen, J.; Siivola, V.; Varjokallio, M.; Arisoy, E.; Saraçlar, M.; Stolcke, A. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. ACM Trans. Speech Lang. Process. 2007, 5, 3. [CrossRef]
- 27. Lai, S.; Liu, K.; He, S.; Zhao, J. How to generate a good word embedding. *IEEE Intell. Syst.* 2016, *31*, 5–14. [CrossRef]
- 28. Maimaiti, M.; Wumaier, A.; Abiderexiti, K.; Wang, L.; Wu, H.; Yibulayin, T. Construction of Uyghur named entity corpus. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan, 7–12 May 2018; p. 14.
- 29. Wang, L.; Wumaier, A.; Maimaiti, M.; Abiderexiti, K.; Yibulayin, T. A semi-supervised approach to Uyghur named entity recognition based on CRF. J. Chin. Inf. Process. 2018, 32, 16–26, 33.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).