Joint Learning Model for Low-Resource Agglutinative Language Morphological Tagging

Gulinigeer Abudouwaili[1,2], **Kahaerjiang Abiderexiti**[1,2], **Nian Yi** [1,2], and **Aishan Wumaier**[1,2,*]
[1]School of Information Science and Engineering, Xinjiang University
[2]Laboratory of Multi-Language Information Technology, Xinjiang University
107556518131@stu.xju.edu.cn, kaharjan@aliyun.com, 15709918429@163.com and Hasan1479@xju.edu.cn

Abstract

Due to the lack of data resources, rule-based or transfer learning is mainly used in the morphological tagging of low-resource languages. However, these methods require expert knowledge, ignore contextual features, and have error propagation. Therefore, we propose a joint morphological tagger for low-resource agglutinative languages to alleviate the above challenges. First, we represent the contextual input with multi-dimensional features of agglutinative words. Second, joint training reduces the direct impact of part-of-speech errors on morphological features and increases the indirect influence between the two types of labels through a fusion mechanism. Finally, our model separately predicts part-of-speech and morphological features. Part-of-speech tagging is regarded as sequence tagging. When predicting morphological features, two-label adjacency graphs are dynamically reconstructed by integrating multilingual global features and monolingual local features. Then, a graph convolution network is used to learn the higher-order intersection of labels. A series of experiments show that the proposed model in this paper is superior to other comparative models.

1 Introduction

Morphological tagging describes the lexical information of a word in a sentence from the part-of-speech (PoS) and morphological features (MFs; case, person, mood, tense, etc.) (Özateş and Çetinoğlu, 2021) and is an essential task in agglutinative language information processing. Morphological tagging can analyze semantics (Klimaszewski and Wróblewska, 2021), so some morphological knowledge will be added to many downstream tasks, such as dependency parsing (Klimaszewski and Wróblewska, 2021), named entity recognition (Kim and Kim, 2020), language models (Park et al., 2021), and machine translation (Jon et al., 2021), to assist the model in learning semantics and improve interpretability. An example is given in Table 1. The first line shows a sentence, the second line shows its lemma, and the third line shows the morphosyntactic description (MSD) labels.

The	cats	are	sleeping	.
the	cat	be	sleep	.
DET	N;PL	V;PRS;3;PL	V;V.PTCP;PRS	PUNT

Table 1: An example of morphological analysis in English

In recent years, there have been many achievements in morphological tagging, among which SIGMORPHON 2019 shared task 2 is a significant milestone (McCarthy et al., 2019), and many cross-lingual morphological tagging models have been proposed. High-resource languages usually use deep learning models and regard morphological tagging as a sequence labeling (Özateş and Çetinoğlu, 2021) or sequence generation task (Oh et al., 2019). The study of English and Chinese morphological tasks began relatively early. Supported by large-scale labeled datasets and large language models, the morphological tagging technology of these languages has reached a mature level. However, the morphological tagging of low-resource languages remains to be further researched. Finite-state transducer (FST) and transfer learning are the primary strategies for constructing morphological tagger in low-resource languages (Wiemerslage et al., 2022; Ibrahim et al., 2018; Rueter et al., 2021; Cotterell and Heigold, 2017). The FST model represents orthographic rules as state transition conditions and can be understood as the transfer of surface relations. Graph convolution networks (GCN) can also explore label relationships (Ma et al., 2021; Zhou et al., 2023). Morphological tagging based on FST focuses on lexical rules of words, which require many linguistic rules. In addition, there are

*:Corresponding author

Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 27–37

problems such as poor semantic ability, ambiguity, rule conflicts, and the inability to express deep lexical rules. In transfer learning, high-resource language knowledge is transferred to a low-resource language, and a tagging model is built by deep learning models. If a sequence labeling or generation model based on deep learning is used with low resources languages, error prediction at the previous time cause error propagation.

In agglutinative languages, a word is formed with a lemma and several suffixes. In the Uyghur word, "almidi" (translation: he/she/they did not pick up), "al" is lemma, "mi(a)", "d" and "i" are suffixes, its MSD label: 'V;SG/PL;3;NEG;PST'. The lemma represents the word's meaning, and the suffix represents the grammatical category (Pan et al., 2020). Each suffix represents grammatical information and corresponds to a morphosyntactic description label (Seker and Tsarfaty, 2020). Morphological taggers in low-resource agglutinative language mainly focus on rule-based and statistical models (Ibrahim et al., 2018; WUSIMAN et al., 2019; Tolegen et al.), while relatively few studies are based on transfer learning or deep learning (Toleu et al., 2017; Liu et al., 2021; Toleu et al., 2022). Conventional methods rely on human-designed rules, which are limited to surface rules in the dataset and cannot capture hidden rules and learn or represent deep grammar rules.

In this paper, we investigate (1) which features in agglutinative language are related to MSD labels, (2) how to reduce the direct impact of error propagation, and (3) whether it is possible to accurately predict more complete MSD labels using label relationships and word representation in low-resource languages. Therefore, to overcome these issues, we first represent the input word by contextual and word-formation features. Second, to reduce error propagation caused by PoS, morphological tagging is divided into PoS tagging and MF tagging. Through literature research and experiments, it has been found that PoS can alleviate ambiguity, and PoS affects the prediction of morphological feature labels. Inspired by the work of (Li et al., 2021), a fusion mechanism is adopted for the middle layer of the two tasks. Finally, the output of the fusion mechanism is input into the conditional random field (CRF) layer to predict the PoS of each word. We pretrain labels in the MF tagging model and calculate relevant label co-occurrence statistics for the high-resource agglutinative language to learn the relationship between labels. The co-occurrence of irrelevant labels is calculated for the Uyghur (Ug), Kazakh (Kz), Tatar (Tt), and Yakut (Yk) datasets. A dynamic adjacency graph is reconstructed by using the above relationships and a GCN to learn the label relationship again to find the hidden relationships between labels. Then, the MF labels of each word are predicted by using the word feature and label relationship. We evaluate our model on four low-resource agglutinative languages, Uyghur, Kazakh, Tatar, and Yakut, in universal dependencies (UD), and experiments show that the performance of the model proposed in this paper is superior to that of other comparable models. The model's average accuracy in four languages reaches 85.29%, and the average F1 score reaches 92.61%.

Our contributions are highlighted as follows:

- This paper proposes a joint morphological tagging model that divides morphological tagging into PoS and MF tagging. Furthermore, the middle layer is fused to transform direct influence into indirect influence.
- To further explore the subtle and hidden relationships between MF labels of low-resource agglutinative languages, this paper describes the universal relationship of agglutinative languages and the characteristic relationship of a monolingual language. The final relationship representation of the monolingual MF labels is dynamically constructed through a GCN.
- We conduct experiments on the Uy, Kz, Tt, and Yk datasets, and the experimental results prove the effectiveness of the model proposed in this paper. This paper also fills the gap in the research on fine-grained morphological tagging of low-resource agglutinative language based on deep learning.

2 Related Work

Morphological processing is the primary task of natural language processing. Relevant tasks include but are not limited to the following: morphological tagging (Özateş and Çetinoğlu, 2021), morphological segmentation (Batsuren et al., 2022), lemmatization (Zalmout and Habash, 2020), and morphological analysis (Wiemerslage et al., 2022). There is also a close connection between these subtasks. For example, morphological analysis can be

split into lemmatization and morphological tagging. Similar to other tasks, morphological tagging tasks can also be summarized as rule learning (Forbes et al., 2021; Kuznetsova and Tyers, 2021), statistical learning (Çöltekin and Barnes, 2019; Mueller et al., 2013), and deep learning(Seker and Tsarfaty, 2020; Li and Girrbach, 2022), according to different research methods. Recurrent neural networks or pretrained language models have been widely used in morphological tagging for high-resource languages. Nine cross-lingual models were submitted for SIGMORPHON 2019 shared task 2, significantly promoting the development of morphological analysis (McCarthy et al., 2019). The winning model, UDify, was proposed by Kondratyuk (2019) and combines a multilingual pretrained language model and several fine-tuning strategies. They trained multilingually over all treebanks in the first stage and then monolingually used saved multilingual weights in the second stage. Finally, the model predicts each grammatical category. Klimaszewski and Wróblewska (2021) proposed a fully neural natural tagging model, COMBO, for accurate PoS tagging, morphological analysis, lemmatization, and (enhanced) dependency parsing. It is a BERT-based end-to-end multilingual model. Li and Girrbach (2022) studied word segmentation and morphological analysis of Sanskrit and proposed three models: word segmentation, morphological analysis, and combined segmentation and analysis models. The combined segmentation and analysis model is an end-to-end pipeline model. Nicolai et al. (2020) proposed a morphological analysis and generation model for more than one thousand languages. They leveraged a parallel corpus to project from English to other low-resource languages and exploited a morphological annotation tool. Two separate sequence transduction models, one neural and one nonneural network model, were trained, and each model produced an N-best list. The tagging model achieved better performance in high resources. Cotterell and Heigold (2017) trained character-level recurrent neural taggers through language transfer to predict the morphological tagger of high and low resource languages. Learning joint character representations among multiple related languages successfully enables knowledge transfer from the high-resource languages to the low-resource, improving accuracy by up to 30%.

It is difficult to achieve high accuracy using a deep learning or cross-lingual transfer learning model for low-resource languages. Because neither FST nor CRF requires a large dataset, it is common to achieve morphological tagging by using these two models for low-resource languages. In the FST method (Tolegen et al.; Kuznetsova and Tyers, 2021), researchers construct language resources, such as morphological and orthographic rules, and then design a morphological analyzer. WUSIMAN et al. (2019) proposed a character-level morphological collaborative analysis model based on the CRF. Morphological segmentation, annotations, and phonetic changes were combined into a composite label for each character. Toleu et al. (2022) proposed a sequence-to-sequence model of morphological disambiguation. It was hypothesized that the vector representation of the correct analysis should be closer to that of the context vector, and the model predicts the correct MSD label by calculating the similarity. FST and statistical methods require many morphological rules or high-quality annotated data, leading to expensive labor costs. Furthermore, the model cannot be combined with contextual features very well, which can easily cause ambiguity problems. In addition, some problems, such as lexical rule conflicts and long running times, may occur.

Therefore, to avoid ambiguity and improve the model performance of predicting more complete MSD labels, the proposed model uses a pretrained model to represent words via contextual features. By jointly training PoS and MF tags, the direct impact of PoS tags on MF tags is reduced. The related morphological tag relations of the agglutinative language in the treebank are learned and transferred to low-resource languages, and introducing irrelevant tag relations between the low-resource languages and dynamically reconstructing the relations between tags, thus alleviating the problem of data resource scarcity.

3 Joint Learning Model for Morphological Tagging

3.1 Task definition

Morphological tagging is a word-level task, and labels are analyzed in terms of context (Wiemerslage et al., 2022). The definition of the morphological tagging task is as follows: a sentence S consists of n words, $S = \{w_1, w_2, \ldots, w_n\}$. The morphological label of the word w_i is T_i, $T_i = \{t_1, t_2, \ldots, t_m\}$, where t_1 is the PoS label, and the following label is the MF label. Therefore, when the input of

the model is sentence S, the output is each word's morphological label, namely, $\{T_1, T_2, \ldots, T_n\}$.

3.2 Model framework

The overall structure of the joint morphological tagging model is shown in Figure 1. We divide the morphological tagging task into two subtasks, PoS tagging and MF tagging. The input sentence is $S = \{\mathbf{E_1}, \mathbf{E_2}, \ldots, \mathbf{E_n}\}$, where $\mathbf{E_i} \in \mathbb{R}^{d_e}$ is the vector of the word w_i, and d_e is the dimension of the vector. First, we input the vector into the BiLSTM layer for each task and encode the sentence through the BiLSTM layer to obtain the hidden state of the context vector, $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n}]$, where $\mathbf{h_i} \in \mathbb{R}^{2d}$. Second, to achieve mutual influence between the two labels, we fuse the output of the BiLSTM layer. Finally, the fused results are input into the inference layer of the model.

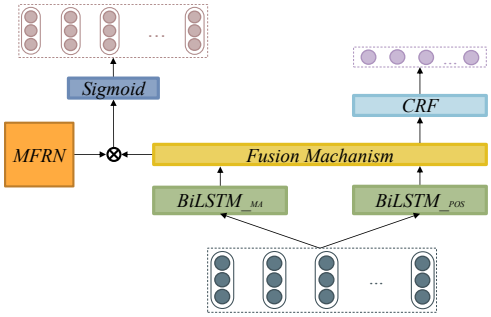

Figure 1: The overall model architecture

The CRF model in the inference layer predicts the PoS label of the word. The morphological feature relation network (MFRN) in the inference layer is dot multiplied by the fused output. We deploy a sigmoid activation function to acquire the probability of each tag, and a tag with a probability value higher than 0.5 is output. The total loss function of the model is:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{PoS} + \lambda\mathcal{L}_{MT} \tag{1}$$

where L_{PoS} is the loss function of the CRF layer, L_{MT} is the loss function of the MFRN network, λ is the manually set weight value, and $\lambda \in [0, 1]$. Through experiments, it is found that when $\lambda = 0.6$, the model performance is the best. The loss function of the PoS tagger is a negative log-likelihood function, as shown in Equation 2. Given N training data, x^i represents the ith input sequence, y^i represents the real tag sequence of the ith sequence, and $\left(y^i \mid x^i\right)$ is the probability of the real tag sequence. The MF tagging loss function is the binary cross-entropy loss function, as shown in Equation 3. Similarly, given N training data, each input sequence has M words, z represents the real label, and $\bar{z}$ represents the label predicted by the model. z^{ijk} indicates whether the $k-th$ label is in the jth word of the ith input sequence.

$$\mathcal{L}_{PoS} = -\sum_{i=1}^{N} \log p\left(y^i \mid x^i\right) \tag{2}$$

$$\mathcal{L}_{\mathcal{MT}} = -\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{|L_{MT}|} z^{ijk} \log \bar{z}^{ijk} + \left(1 - z^{ijk}\right) \log\left(1 - \bar{z}^{ijk}\right) \tag{3}$$

3.2.1 Input embedding layer

The input embedding $\mathbf{E_i}$ consists of three parts: word embedding $\mathbf{W_i}$, morphological embedding $\mathbf{M_i}$ and local context embedding $\mathbf{C_i}$. The dimension of each embedding is d_e. The embeddings of words and local context are generated by the pretrained language model. We use the Chinese minority pretrained language model[1] (CINO) for Uyghur and the pretrained cross-lingual language model[2] (XLM)-RoBERTa for Kazakh, Tatar and Yakut. Since the pretrained models break the morphological rules by splitting words into different subwords, we use a BiLSTM layer to generate Ug and Kz morphological embeddings (Abuduwaili et al., 2022; Makhambetov et al., 2015) (Tt and Yk have no available morphological segmentation tools, and no morphological embeddings are added, only char-based embeddings are used). The final input is generated by concatenating the word, local context and morphological embeddings. The structure of the input layer is shown in Figure 2.

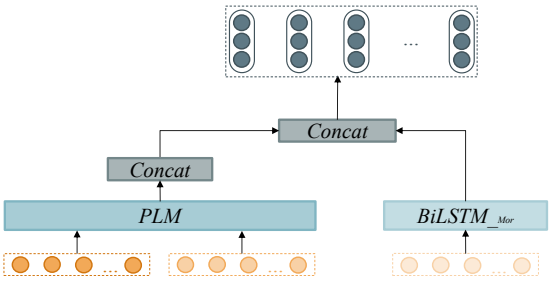

Figure 2: Input layer structure

[1]https://huggingface.co/hfl/cino-base-v2
[2]https://huggingface.co/xlm-roberta-base

3.2.2 Fusion mechanism

We design a fusion mechanism to effectively exchange information between the $\text{BiLSTM}_{\text{PoS}}$ and $\text{BiLSTM}_{\text{MF}}$ layers. The output results of the $\text{BiLSTM}_{\text{PoS}}$ layer, where $\mathbf{H_{PoS}} = [\mathbf{h_1^{PoS}}, \mathbf{h_2^{PoS}}, ..., \mathbf{h_n^{PoS}}]$, and $\text{BiLSTM}_{\text{MF}}$ layer, where $\mathbf{H_{MF}} = [\mathbf{h_1^{MF}}, \mathbf{h_2^{MF}}, ..., \mathbf{h_n^{MF}}]$, are nonlinearly transformed to generate $\mathbf{H_{PoS}^{new}}$ and $\mathbf{H_{MF}^{new}}$, and $\sigma(\cdot)$ denotes the softmax activation function, as follows:

$$\mathbf{H}_{MF}^{new} = \sigma\left((\mathbf{W}_{MF}\mathbf{H}_{MF}) \times \mathbf{H}_{PoS}^{T}\right) \times \mathbf{H}_{PoS} \tag{4}$$

$$\mathbf{H}_{PoS}^{new} = \sigma\left((\mathbf{W}_{PoS}\mathbf{H}_{PoS}) \times \mathbf{H}_{MF}^{T}\right) \times \mathbf{H}_{MF} \tag{5}$$

3.2.3 Morphological feature relation network

The input of this module consists of three parts: the relevant label adjacency (RLA) matrix, irrelevant label adjacency (ILA) matrix, and label embedding (LE). A detailed description of the label relation is provided in the Appendix A. In this paper, we construct an MF label dataset using agglutinative languages for UD datasets. We utilize the label's co-occurrence to obtain the label embedding in this dataset. Given the label set $L = \{l_1, l_2, ..., l_n\}$, the label embedding after training is $\mathbf{E}_l \in \mathbb{R}^{|n| \times d_l}$, where n is the number of label types and d_l is the embedding dimension. Each label embedding is:

$$\mathbf{e}_l^i = \mathbf{E}_l(l_i) \tag{6}$$

The initial label input of the MFRN is the pretrained label embedding, which is finetuned during the training. The structure of the MFRN is shown in Figure 3.

The two adjacency matrices represent the relationships between relevant and irrelevant labels. Due to the lack of morphological tagging datasets in both languages, we construct a label adjacency matrix to explore the universal relationship between labels more thoroughly using the above labeled datasets. Each language has characteristic features that affect the relationships between related labels. Therefore, four irrelevant label adjacency matrices are constructed for the Ug, Kz, Tt and Yk datasets. The relevant label adjacency matrix is constructed on all languages, which has universal features, while the irrelevant label adjacency matrices have unique features for each language.

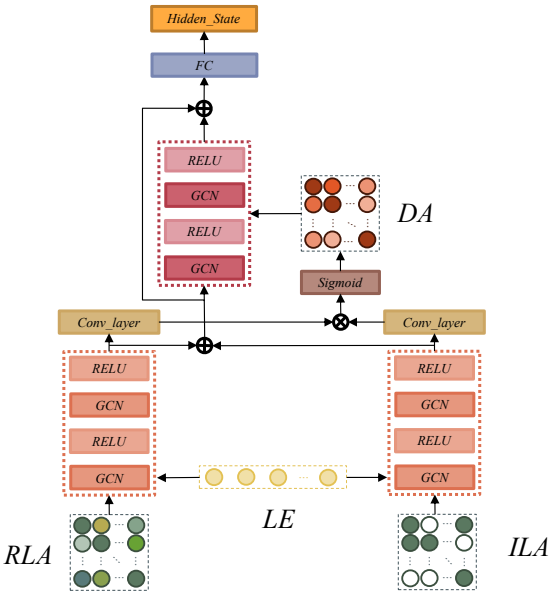

Figure 3: The MFRN structure

Then, we utilize a GCN (Kipf and Welling, 2016) to learn the different relationships between labels guided by the label adjacency matrices. Therefore, labels are treated as nodes and label relationships as edges, and morphological relationships are projected into undirected graphs. An undirected graph $\mathfrak{g}$ with n nodes is represented by an adjacency matrix, $\mathbf{RLA} \in \mathbb{R}^{n \times n}$. Similarly, $\mathbf{ILA} \in \mathbb{R}^{n \times n}$. Each element $\mathbf{A}_{ij}$ in the matrix indicates the relationship between the ith and the jth nodes. Specifically, $\mathbf{ILA}_{ij} = 1$ if the ith node is connected to the jth node, and $\mathbf{ILA}_{ij} = 0$ otherwise. After that, the two matrices are normalized as follows:

$$\widetilde{\mathbf{RLA}} = \mathbf{D}_{RLA}^{-\frac{1}{2}}\overline{\mathbf{RLA}}\mathbf{D}_{RLA}^{-\frac{1}{2}} \tag{7}$$

$$\widetilde{\mathbf{ILA}} = \mathbf{D}_{ILA}^{-\frac{1}{2}}\overline{\mathbf{ILA}}\mathbf{D}_{ILA}^{-\frac{1}{2}} \tag{8}$$

where $\mathbf{D}_{RLA}$ and $\mathbf{D}_{ILA}$ are diagonal degree matrices with entries $\mathbf{D}_{RLA_{ij}} = \sum_j \mathbf{RLA}_{ij}$, and $\mathbf{D}_{ILA_{ij}} = \sum_j \mathbf{ILA}_{ij}$, respectively. We add a self-loop for each node in the GCN. $\overline{\mathbf{ILA}} = \mathbf{ILA} + \mathbf{I}_N$ and $\overline{\mathbf{RLA}} = \mathbf{RLA} + \mathbf{I}_N$, where $\mathbf{I}_N$ is the identity matrix. The multilayer GCN learns the relationship between labels. The output of layer $\mathbf{H}^l$ is:

$$\mathbf{H}^l = \sigma\left(\bar{\mathbf{A}}\mathbf{H}^{l-1}\mathbf{W}^{l-1}\right) \tag{9}$$

where $\sigma(\cdot)$ denotes the leaky rectified linear unit activation function, $\bar{\mathbf{A}}$ represents the normalized adjacency matrix ($\overline{\mathbf{RLA}}$ and $\overline{\mathbf{ILA}}$), $\mathbf{H}^{l-1}$ represents the output of the previous layer, and $\mathbf{W} \in \mathbb{R}^{d_l \times d'}$

represents the parameter to be learned. $\mathbf{H}^0$ is the initial label embedding. The final outputs of the two multilayer GCNs are concatenated to represent the new label embedding, as follows:

$$\mathbf{LE} = \mathbf{H}^2_{RL} \oplus \mathbf{H}^2_{IL} \tag{10}$$

where $\mathbf{H}^2_{RL}$ and $\mathbf{H}^2_{IL}$ represent the output of the relevant label and irrelevant label fed into the multilayer GCN, respectively, and the new label embedding $\mathbf{LE} \in \mathbb{R}^{|n| \times 2d'}$. It is difficult to find the hidden relationship between labels only by the statistical rules of labels in the training data, and there will also be noise. Therefore, we utilize the GCN to extract the features from $\mathbf{H}^2_{RL}$ and $\mathbf{H}^2_{IL}$, and multiply the two results to dynamically reconstruct the adjacency graph $\mathbf{DA}$:

$$\mathbf{DA} = \sigma\left(\left(\mathbf{W}_a\mathbf{H}^2_{RL}\right) \times \left(-\mathbf{W}_b\mathbf{H}^2_{IL}\right)^T\right) \tag{11}$$

where $\mathbf{W}_a$ and $\mathbf{W}_b$ are 1×1 convolution layers, and $\sigma(\cdot)$ denotes the sigmoid activation function. We normalize the reconstruction adjacency matrix and obtain a new dynamic reconstruction adjacency matrix $\widetilde{\mathbf{DA}}$. The newly generated label embedding and adjacency matrix are input into another multilayer GCN, and the output result of the dynamic reconstruction network $\mathbf{H}^4$ is obtained. Finally, we combine the output of each GCN layer through a fully connected layer to output the final label representation $\mathbf{HS}$, where $\mathbf{H}^F \in \mathbb{R}^{|n| \times 3d'}$, as follows:

$$\mathbf{H}^F = \mathbf{H}^4 \oplus \mathbf{H}^2_{RL} \oplus \mathbf{H}^2_{IL} \tag{12}$$

$$\mathbf{HS} = \mathbf{W}_c\mathbf{H}^F \tag{13}$$

4 Experiments

4.1 Data

We evaluate the proposed model with low-resource agglutinative languages in the UD treebank (version 2.10), such as the Uyghur, Kazakh, Tatar, and Yakut (UKTY) datasets[3]. The UD treebank is a sentence-level dataset that includes many types of PoS labels. Since the annotation type in UD is in the CoNLL-U format, to ensure the integrity of morphological labels and the comparability of experimental results, we use a tool (McCarthy et al., 2018) to convert the data in CoNLL-U format into the UniMorph format and split all datasets into training, testing, and validation sets in an 8:1:1 proportion. The statistics of the dataset are shown in Table 2.

[3]https://universaldependencies.org/#download

4.2 Experimental results and analysis

4.2.1 Overall performance

We seek to compare our model to recurrent cross-lingual models with the best performance in this experiment. The experimental results are shown in Table 3. A detailed description of implementation details is provided in Appendix B.

Neural tagger (McCarthy et al., 2019): This is the baseline model of SIGMORPHON 2019 shared task 2. It generates the morphological tag sequence of each word through a multilayer BiLSTM model. **Multi-Team**: Üstün et al. (2019) proposed a multi-team (multi-attention and multi-decoder) morphological analysis model. The model's performance is improved by introducing pretrained word embeddings to initialize the input. **Morpheus** (Yildiz and Tantuğ, 2019): This model generates word embedding and context-aware word embedding using LSTM and BiLSTM and then inputs the two kinds of word embedding into a decoder to generate morphological labels. **UDify**: Kondratyuk (2019) proposed a morphological tagging model based on multilingual bidirectional encoder representations from transformers (BERT). This model was the winner of SIGMORPHON 2019 shared task 2. **UDPipe**: This model proposed by Straka and Straková (2020) adds pretrained contextualized embeddings (generated by BERT) to the input and uses individual morphological features for regularization. UDPipe placed second in SIGMORPHON 2019 shared task 2. **COMO**: Klimaszewski and Wróblewska (2021) proposed a fully neural NLP tagging system of PoS, morphological analysis, lemmatization, and dependency parsing. It achieved better prediction quality than that of SOTA methods at the time.

Table 3 shows the experimental results of several SOTA models and our model for the UKTY datasets. We measure the performance of models in terms of precision (P), recall (R), F1 score, and accuracy (ACC). The model proposed in this paper achieves a high F1 score and accuracy. Because UDify, UDPipe, and COMBO models are based on BERT, we also conducted comparative experiments on BERT. The experimental results of these models show that except in the Tt, the accuracy and F1 score are not significantly different, and the F1 score is slightly lower. This is because MSD is a set of labels, and the accuracy evaluates the integrity of the predicted MSD label, while the F1 score (P and R) evaluates whether the prediction results for each

Dataset	UD_Uyghur-UDT				UD_Kazakh-KTB				UD_Tatar-NMCTT				UD_Yakut-YKTDT			
	Train	Valid	Test	Label	Train	Valid	Test	Label	Train	Valid	Test	Label	Train	Valid	Test	Label
#Sentences	2764	346	346	45	862	108	108	57	118	15	15	50	231	29	29	36
#Words	32174	3538	4615		8483	1085	1121		1776	222	282		1144	131	128	

Table 2: Dataset statistics.

Lang	Ug				Kz				Tt				Yk				Avg.	
	P	R	F1	ACC.	P	R	F1	ACC.	P	R	F1	ACC.	P	R	F1	ACC.	F1	ACC.
Neural tagger	85.90	87.30	86.60	79.40	87.39	86.86	87.12	78.31	**84.77**	**83.22**	**83.99**	**75.80**	**88.31**	**85.80**	**87.04**	**84.38**	86.19	79.47
Multi-Team	89.56	**89.86**	89.71	82.54	88.91	88.71	88.81	80.38	79.16	79.72	79.44	65.25	84.38	82.09	83.22	77.42	85.30	76.40
Morpheus	**90.38**	89.09	**89.73**	**83.90**	88.39	86.82	87.60	78.50	59.27	54.93	57.02	39.01	70.07	67.19	68.60	62.50	75.74	65.98
UDify	88.42	88.16	88.29	83.73	**92.79**	**92.52**	**92.66**	**86.98**	83.38	82.50	82.94	75.53	85.53	84.94	85.25	78.91	87.29	81.29
UDPipe*	88.15	89.43	88.79	81.13	-	-	-	-	-	-	-	-	-	-	-	-	88.79	81.13
COMBO**	86.48	86.07	86.27	77.14	-	-	-	-	-	-	-	-	-	-	-	-	86.27	77.14
Our model-BERT	90.87	87.05	88.92	81.05	95.24	95.16	95.20	88.40	85.27	83.38	84.31	65.38	87.58	86.95	87.26	77.31	88.86	76.73
Our model-XLM/CINO	**94.87**	**96.18**	**95.52**	**91.60**	**96.92**	**97.32**	**97.12**	**92.82**	**89.18**	**87.20**	**88.18**	**76.40**	**90.03**	**89.23**	**89.63**	80.34	**92.61**	**85.29**

* The UDPipe web service can be accessed directly via API. And only the Uyghur model is available, so only Uyghur is tested in the test set.

** The COMBO model is only available on Uyghur, and the trained model is loaded in the test.

Table 3: Experimental results of morphological tagging.

label in the MSD label are correct. In languages with extremely low resources (Yk and Tt), neural tagger is superior to other baseline models. UDify demonstrated stable performance in four languages. For the Ug dataset, compared with the results of Morpheus, our CINO-based model increases the F1 score and accuracy by 5.79% and 7.70%, respectively. For the Kz dataset, compared with the results of UDify, our XLM-based model increases the F1 score and accuracy by 4.46% and 5.84%, respectively. For the Tt dataset, compared with the results of neural tagger, our XLM-based model increases the F1 score and accuracy by 4.19% and 0.60%, respectively. For the Yk dataset, compared with the results of neural tagger, our XLM-based model increases the F1 score by 2.59% and decreases the accuracy by 4.04%.

4.2.2 Analysis and discussion

To further verify the impact of each module on the model performance, we perform a series of experiments. The experiments mainly explore the influence of the input, fusion mechanism, and different label relationships on the model performance.

Different inputs. Morphology and context will influence the morphological labels of words. Therefore, to explore the impact of different input features on the model performance, a group of ablation experiments is conducted around words, morphology, and local context, and the experimental results are shown in Table 4.

From the experimental results in Table 4, it is found that local context and morphology can affect morphological tagging. Adding local context in low-resource agglutinative language can increase

Lang.	w	w+3g	w+5g	w+7g	w+m	w+m+3g
Ug	86.64	89.32	90.31	90.45	88.95	91.60
Kz	86.79	91.17	90.56	91.08	90.24	92.82
Tt	75.80	76.40	76.80	79.12	-	-
Yk	75.31	80.34	74.53	62.34	-	-

Table 4: The impact of different inputs on model accuracy (%). 'w' represents word, 'g' represents gram, and 'm' represents morphology.

the model's accuracy, but sentence length can affect the growth rate. Because the sentence is shorter, the model learns less critical contextual information and more noise. Based on Table 2, it is found that the average sentence level of the Yk dataset is less than 5 (the average length of sentence is 4.41), so when the window size is 5-7, the model's accuracy starts to drop. Therefore, considering the dataset and practical application, this model uses the local context feature with a window size of 3 in the model. Words are composed of lemma and suffixes (morpheme) in agglutinative languages. The suffixes deeply affect morphological labels. In Table 4, adding morphological features to the Ug and Kz languages can also significantly improve the accuracy of the model.

Fusion mechanism. We jointly train the PoS and MF tagging models to reduce the impact of error propagation. However, the impact of the two labels on each other cannot be ignored in morphological tagging. Table 5 shows the experimental results without a fusion mechanism:

Compared to the four datasets, the fusion mechanism performs more significantly on datasets with smaller datasets (Tt and Yk). From the experimental results, it was found that the average F1 score

Lang.	F1	ACC.
Ug	94.05 (↓1.47)	89.64(↓1.96)
Kz	96.11(↓1.01)	91.10(↓1.72)
Tt	80.07(↓8.11)	73.67(↓2.73)
Yk	78.46(↓11.17)	63.91(↓16.43)
Avg.	↓Δ5.44	↓Δ5.71

Table 5: The influence of the fusion mechanism on the model.

and average accuracy of the model with the fusion mechanism increased by 5.44% and 5.71%, respectively. It also proves that the two labels have an interactive relationship, and the fusion mechanism also plays an indirect constraint role.

Different label relationships. We conduct ablation experiments for adding initial label (INL) relationships, irrelevant label (IRL) relationships, pretrained label (PL) relationships and reconstructed label (RL) relationships in the model. The experimental results are shown in Table 6.

Relation	Ug	Kz	Tt	Yk
baseline	86.75	88.01	70.53	72.66
baseline + INL	85.61	70.05	39.33	42.72
baseline+INL+IRL	88.62	88.91	73.97	75.44
baseline+PL+IRL	89.97	91.01	75.67	78.99
baseline+PL+IRL+RL	91.60	92.82	76.40	80.34

Table 6: The influence of different label relationships. We report accuracy (%) of MSD label for all tokens.

In Table 6, the baseline represents the model without any relationship learning. The baseline learns and represents the relationship between the surface and the inside of the label by adding different label relationships and finally outputs the MF label of the word in combination with the input content. In the experiment, when adding the initial label relationship (statistical relationship), the accuracy of the model in Tt and Yk significantly decreases. After adding irrelevant label relationships, the accuracy rate shows an upward trend. Although prior knowledge was added to the model, it was difficult to fully learn the relationships between labels due to the small dataset with various labels. After adding prior knowledge of irrelevant labels, it can constrain the relationships between labels. We believe that there are similarities between languages of the same language family. After adding pretrained label embedding, the model learns the universal relationship of the agglutinative MF labels. Adding the reconstruction relationship enables the model to capture more label relationships in dynamic learning. These methods can learn the hidden relationship between labels, thus improving the robustness of the model.

Error analysis. This paper selects Kazakh, which has the more types of morphological labels (57) and has the longest morphological labels (8) in the dataset, as an example to further analyze the model's performance in predicting long labels compared to other models. The model proposed in this paper can effectively predict short- or long-label problems through label relationships. The experimental results are shown in Table 7.

Label length	1-3	4	5	6	7	8
Overlap	53.82	23.53	20.00	56.00	54.35	57.14
Neural tagger	20.21	29.41	60.00	28.00	32.61	21.43
Multi-Team	17.99	27.45	60.00	26.00	30.43	28.57
Morpheus	18.74	35.29	80.00	30.00	39.13	50.00
UDify	9.84	37.25	60.00	30.00	26.09	21.43
Our model	5.82	13.73	40.00	18.00	13.04	7.14

Table 7: The impact of label length and both (word and lemma) overlaps (%) on model error rate (%). Both overlap means the word and lemma occur in the training and test set.

From Table 7, it is found that in addition to the impact of data size on model performance, the length of labels and the overlap can also affect model performance. The label length is short (1-3), and the overlap is high, so the model's error rate is also relatively low. When the label length is 5, the overlap is the lowest, and the model's error rate is the highest. When the overlap is approximately similar (length=1-3 and length=7, Δ=0.53%), the label length is longer, and the model's error rate is higher. Compared with a label length of 6 and 8, although the overlap is slightly different (Δ=1.14%), the neural tagger, UDify, and our model have lower errors when the label length is 8. This is the result of overlap influencing the model. Compared to other models, our model is less affected by label length and overlap.

5 Conclusion

This paper proposes a joint morphological tagging model based on a neural network for low-resource agglutinative languages. First, to effectively capture the multi-dimensional information of the input words, the model uses the morphological, word and local context features to represent the input words. Second, to reduce the impact of the PoS label on the MF label, the two models are jointly trained, and a fusion mechanism is used to complete the interac-

tion between the two middle layers. Finally, PoS tagging is regarded as sequence labeling, and the CRF model predicts the PoS tag. MF tagging is regarded as a classification task, and a new adjacency graph is dynamically reconstructed by using the two relationships between labels and GCNs. GCNs are used to express the higher-order relationship between labels. This model combines the universal features of agglutinative languages and the characteristic features of UKTY languages, learns the relationship between labels, and effectively alleviates the problems caused by data scarcity. The experiments conducted with UD treebanks show that the proposed morphological tagging model outperforms other models. We explore the morphological tagging model based on a neural network under low resources as an example. In future research, we will continue to optimize the model's relationship representation and threshold selection abilities, and further improving the model's performance.

Limitations

The method proposed in this paper has some limitations:

(1) This model learns not only the features of words, word formation, and context but also the relationships between labels. When there is too much noise, it can disturb the relationship between labels.

(2) The model needs more training time than other models to learn pretrained label relationships.

Acknowledgements

We gratefully thank the anonymous reviewers for their insightful comments. This work is supported by the National Natural Science Foundation of China (grant numbers 62166044, 61762084) and the Natural Science Foundation of Xinjiang Uyghur Autonomous Region under Grant No.2021D01C079.

References

Gulinigeer Abuduwaili, Kahaerjing Abiderexiti, Yunfei Shen, and Aishan Wumaier. 2022. Research on the uyghur morphological segmentation model with an attention mechanism. *Connection Science*, 34:2577–2596.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Çağrı Çöltekin and Jeremy Barnes. 2019. Neural and linear pipeline approaches to cross-lingual morphological analysis. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 153–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Clarissa Forbes, Garrett Nicolai, and Miikka Silfverberg. 2021. An FST morphological analyzer for the gitksan language. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 188–197, Online. Association for Computational Linguistics.

Tuergun Ibrahim, Kahaerjiang Abiderexiti, Aishan Wumaier, and Maihemuti Maimaiti. 2018. A survey of central asian language processing. *Journal of Chinese Information Processing*, 32(5):14.

Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. End-to-end lexically constrained machine translation for morphologically rich languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4019–4033, Online. Association for Computational Linguistics.

Hongjin Kim and Harksoo Kim. 2020. Integrated model for morphological analysis and named entity recognition based on label attention networks in korean. *Applied Sciences*, 10(11):3740.

Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv e-prints*, page arXiv:1609.02907.

Mateusz Klimaszewski and Alina Wróblewska. 2021. COMBO: State-of-the-art morphosyntactic analysis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 50–62, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Kondratyuk. 2019. Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics,*

Phonology, and Morphology, pages 12–18, Florence, Italy. Association for Computational Linguistics.

Anastasia Kuznetsova and Francis Tyers. 2021. A finite-state morphological analyser for Paraguayan Guaraní. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 81–89, Online. Association for Computational Linguistics.

Jingwen Li and Leander Girrbach. 2022. Word Segmentation and Morphological Parsing for Sanskrit. *arXiv e-prints*, page arXiv:2201.12833.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6319–6329, Online. Association for Computational Linguistics.

Chang Liu, Abudoukelilu ABULIZI, Dengfeng YAO, and Halidanmu ABUDUKELIMU. 2021. Survey for uyghur morphological analysis. *Computer Engineering and Applications*, 57(15):42–61.

Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. 2021. Label-specific dual graph neural network for multi-label text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3855–3864, Online. Association for Computational Linguistics.

Olzhas Makhambetov, Aibek Makazhanov, Islam Sabyrgaliyev, and Zhandos Yessenbayev. 2015. Data-driven morphological analysis and disambiguation for kazakh. In *Computational Linguistics and Intelligent Text Processing*, pages 151–163, Cham. Springer International Publishing.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

Garrett Nicolai, Dylan Lewis, Arya D. McCarthy, Aaron Mueller, Winston Wu, and David Yarowsky. 2020. Fine-grained morphosyntactic analysis and generation tools for more than one thousand languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3963–3972, Marseille, France. European Language Resources Association.

Byung-Doh Oh, Pranav Maneriker, and Nanjiang Jiang. 2019. THOMAS: The hegemonic OSU morphological analyzer using seq2seq. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 80–86, Florence, Italy. Association for Computational Linguistics.

Şaziye Betül Özateş and Özlem Çetinoğlu. 2021. A language-aware approach to code-switched morphological tagging. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 72–83, Online. Association for Computational Linguistics.

Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Multi-task neural model for agglutinative language translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 103–110, Online. Association for Computational Linguistics.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Jack Rueter, Niko Partanen, Mika Hämäläinen, and Trond Trosterud. 2021. Overview of open-source morphology development for the Komi-Zyrian language: Past and future. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*, pages 29–39, Syktyvkar, Russia (Online). Association for Computational Linguistics.

Amit Seker and Reut Tsarfaty. 2020. A pointer network architecture for joint morphological segmentation and tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4368–4378, Online. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2020. UDPipe at EvaLatin 2020: Contextualized embeddings and treebank embeddings. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 124–129, Marseille, France. European Language Resources Association (ELRA).

G. Tolegen, A. Toleu, and R. Mussabayev. A finite state transducer based morphological analyzer for the kazakh language. In *2022 7th International Conference on Computer Science and Engineering (UBMK)*, pages 01–06.

Alymzhan Toleu, Gulmira Tolegen, and Aibek Makazhanov. 2017. Character-aware neural morphological disambiguation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 666–671, Vancouver, Canada. Association for Computational Linguistics.

Alymzhan Toleu, Gulmira Tolegen, and Rustam Mussabayev. 2022. Language-independent approach for morphological disambiguation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5288–5297, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ahmet Üstün, Rob van der Goot, Gosse Bouma, and Gertjan van Noord. 2019. Multi-team: A multi-attention, multi-decoder approach to morphological analysis. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 35–49, Florence, Italy. Association for Computational Linguistics.

Adam Wiemerslage, Miikka Silfverberg, Changbing Yang, Arya McCarthy, Garrett Nicolai, Eliana Colunga, and Katharina Kann. 2022. Morphological processing of low-resource languages: Where we are and what's next. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 988–1007, Dublin, Ireland. Association for Computational Linguistics.

Tuergong WUSIMAN, YaTing Yang, Aizizi TUERXUN, and Li CHENG. 2019. Collaborative analysis of uyghur morphology based on character level. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 55(01):47–54.

Eray Yildiz and A. Cüneyd Tantuğ. 2019. Morpheus: A neural network for jointly learning contextual lemmatization and morphological tagging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 25–34, Florence, Italy. Association for Computational Linguistics.

Nasser Zalmout and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online. Association for Computational Linguistics.

Xiaotang Zhou, Tao Zhang, Chao Cheng, and Shinan Song. 2023. Dynamic multichannel fusion mechanism based on a graph attention network and bert for aspect-based sentiment classification. *Applied Intelligence*, 53:6800–6813.

A Label Relationship

We divide label relationships into two categories: relevant and irrelevant relationships. As shown in the label adjacency matrix in Table 8, if there are nonzero data in the relative position of two labels, they are relevant; otherwise, they are irrelevant. For example, labels A and B are relevant, and labels A and D are irrelevant.

	A	B	C	D
A	0	5	10	0
B	5	0	4	0
C	10	4	0	0
D	0	0	0	0

Table 8: Label adjacency matrix

B Implementation Details

The experimental environment is based on Python 3.8 [4] and the PyTorch 1.9.0 deep learning framework [5]. The word vector dimension is 768, the morphological embedding dimension is 300, the number of $\mathrm{BiLSTM}_{\mathrm{mor}}$ hidden units is 768, and the local context vector dimension is 768. $\mathrm{BiLSTM}_{\mathrm{PoS}}$ and $\mathrm{BiLSTM}_{\mathrm{MT}}$ each have 768 hidden units, and the label embedding dimension is 300. The Adam optimizer is used for training, and a dropout rate of 0.5 is enforced during training. We train each configuration using a batch size of 64, a learning rate of 0.01, and the leaky rectified linear unit activation function in the GCN.

[4] https://www.python.org/downloads/
[5] https://pytorch.org/