An End-to-End Scalable Iterative Sequence Tagging with Multi-Task Learning

Lin Gui[1,2], Jiachen Du[1], Zhishan Zhao[3], Yulan He[2], Ruifeng Xu[1(✉)], and Chuang Fan[1]

[1] Harbin Institute of Technology (Shenzhen), Shenzhen, China
xuruifeng@hit.edu.cn
[2] Aston University, Birmingham, UK
[3] Baidu Inc., Beijing, China

Abstract. Multi-task learning (MTL) models, which pool examples arisen out of several tasks, have achieved remarkable results in language processing. However, multi-task learning is not always effective when compared with the single-task methods in sequence tagging. One possible reason is that existing methods to multi-task sequence tagging often reply on lower layer parameter sharing to connect different tasks. The lack of interactions between different tasks results in limited performance improvement. In this paper, we propose a novel multi-task learning architecture which could iteratively utilize the prediction results of each task explicitly. We train our model for part-of-speech (POS) tagging, chunking and named entity recognition (NER) tasks simultaneously. Experimental results show that without any task-specific features, our model obtains the state-of-the-art performance on both chunking and NER.

Keywords: Multi-task learning · Interactions · Sequence tagging

1 Introduction

Sequence tagging is one of the most important topics in Natural Language Processing (NLP), encompassing tasks such as part-of-speech tagging (POS), chunking, and named entity recognition (NER). In recently years, neural network (NN) based models have achieved impressive results on various sequence tagging tasks, including POS tagging [1,2], chunking [3,4], and NER [5,6].

One of the challenges for sequence tagging tasks is that there is not enough training data to train a good model. Heavy handcrafted features and language-specific knowledge resources are costly to develop in new sequence tagging tasks [5]. To overcome this problem, multi-task learning (MTL) models have been proposed.

MTL is an important mechanism that aims to improve the generalization of model performance by learning a task together with other related tasks [7]. Several NN based MLT models have been applied to various sequence tagging tasks

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 288–298, 2018.
https://doi.org/10.1007/978-3-319-99501-4_25

[4,8,9]. These models use shared representations constructed by some lower layers, then stack several task-specific upper layers to learn task-relevant representations. However, representation sharing in lower NN layers only captures weak interactions between different tasks. That perhaps one reason of that multi-task learning is not always effective compared with single-task methods in sequence tagging [10].

To improve MTL performance, we propose a unified multi-task learning framework for sequence tagging which could iterative utilize predicted tags of each task explicitly. Essentially, our model predicts tags of all tasks with several iterations. The learned distributions of task-specific prediction in every iteration are merged with the shared representations for tag sequence prediction in the next iteration. The iterative training mechanism gives the model capability to refine prediction results of one task by simultaneously considering prediction results of other tasks.

In particular, we propose a CNNs-Highway-BLSTM as base model for sequence tagging. Character-level and word-level convolutional neural networks (CNNs) are used to capture morphological features and a Highway network is implemented to keep valuable features by an adaptive gating units. These features are fed into a Bidirectional Long Short-term Memory Network (BLSTM) for task-specific prediction. According to our iterative training mechanism, the distribution of prediction can be concatenated with shared representations from Highway layer output and sent into BLSTM in the next iteration. The experimental results show that our base model outperforms the state-of-the-art methods on Chunking and NER with iterative training process.
The main contributions of this paper are:

- We propose a new framework to explicitly make use of predicted tags from one task as additional features for other tasks and hence better capture interactions between different tasks in multi-task learning.
- We proposed a CNNs-Highway-BLSTM as a base model for sequence tagging.
- We evaluate our model on several benchmarking datasets and achieve the state-of-art performance on both chunking and NER.

2 Our Approach

In this section, we will first describe a general framework of multi-task learning for sequence tagging and introduce our idea of iterative training. We will then propose a base sequence tagging model which includes lower layers of character-level CNN, word-level-CNN, Highway network and BLSTM that shared across all tasks. At the end of this section, we will present the details of model training.

2.1 Multi-task Learning

Multi-task Learning (MTL) aims to improve model generalization by learning multiple tasks simultaneously from shared representations based on the assumption that features learned for each task can be helpful for other tasks. In addition,

pooling samples of several tasks could also potentially generalize model well [7]. As we focus on sequence tagging here, both inputs and outputs are in the vector form. Given an input sequence $\boldsymbol{x}$, the MLT model outputs a tagging sequence $\overrightarrow{y^{(i)}}$ for different tasks. The model typically consists of two kinds of parameters: task-specific parameters $h^{(i)}$, which are in the upper layer of the architecture shown in Fig. 1, and the parameters $h^{(shared)}$ shared across all tasks, which are in the lower layer of the architecture shown in Fig. 1.

In MTL, it is reasonable to assume that tags predicted for one task can be useful features for other tasks. For instance, a word with POS tag "noun" is more likely to be a named entity compared to others tagged as cardinal numeral. Conventionally, lots of NLP systems use features obtained from the output of other preexisting NLP systems [4].

In order to explicitly use the tag prediction results from one task in others in MTL, we propose an iterative training procedure for MTL as shown in Fig. 1, in which steps enclosed in a dashed box are repeated for a number of iterations. The inputs to each iteration consist of:

1. $h^{(shared)}$, features extracted from data by a shared layer;
2. $\boldsymbol{y} = [\overrightarrow{y^{(1)}}, \overrightarrow{y^{(2)}}]$, concatenated prediction probabilities of all tasks from the previous iteration.

Both inputs are concatenated and fed to BLSTM. It is worth noting that BLSTM is shared across all tasks and it is separated from $h^{(shared)}$ because it participates in all iterations. Our proposed iterative training procedure (Fig. 1) allows the model to incorporate the predicted results of all tasks as additional features in the next iteration. With BLSTM, the model extends tags interaction to the sentence level. To ensure that the predicted results at each iteration is close to the true tag sequence, we define a cost function by taking into account the differences between the predicted and true tag sequences in all iterations and for all tasks:

$$cost = \frac{1}{T}\sum_{i=1}^{T} L(\overrightarrow{y_t}, \overrightarrow{y*}), \tag{1}$$

$$L(\overrightarrow{y_t}, \overrightarrow{y*}) = \frac{1}{M}\sum_{m=1}^{M} \alpha_m \tilde{L}(\overrightarrow{y_t^{(m)}}, \overrightarrow{y^{(m)*}}) \tag{2}$$

where $\overrightarrow{y_t^{(m)}}$ is the prediction probabilities sequence of task m at iteration t, $\overrightarrow{y^{(m)*}}$ is the ground-truth label sequence of task m, T is the number of iterations, M is the number of tasks, α_m is the weight of different tasks, $\tilde{L}$ is the cross entropy function. For task m, the final prediction result is a mean of predictions across all iterations:

$$\overrightarrow{y^{(m)}} = \frac{1}{T}\sum_{i=1}^{T} \overrightarrow{y_t^{(m)}} \tag{3}$$

2.2 Basic Sequence Tagging Model

In the previous subsection, we have introduced the iterative training procedure and the cost function used in MTL. Since the predicted tag sequence will be used as additional features in next iteration of model training, a strong base sequence tagging model is needed for obtaining better performance. In this section, we will propose a NN-based sequence tagging model, which is called CNNs-Highway-BLSTM. It starts with character-level and word-level CNNs to capture morphological and contextual features. Next, a Highway network is used to keep valuable features across word-level CNNs by using *transform* and *carry* gates. These features are input into a BLSTM for capturing sequence long-term dependencies. The output of BLSTM is fed into different task-specific output layers.

Word-Level CNN. The word-level CNN takes an input sentence, called *Word Representation*, as a sequence of words $x = [x_1, x_2, ..., x_n]$ where each word is represented as a d-dimensional vector, and returns another sequence $S = [s_1, s_2, ..., s_n]$ which represents local information about the sequence at every word of the input. A narrow convolution is applied between x and a kernel $W \in \mathbb{R}^{kd}$ of width k. $\lfloor \frac{k}{2} \rfloor$ and $\lfloor \frac{k-1}{2} \rfloor$ padding vectors are added to the head and tail of sequence to make sure the sequence length does not change after the convolution layer.

Character-Level CNN. Character-level CNNs have been shown to be an effective method to extract morphological features from characters of words [2,11]. Given a word, we first apply its character embedding to a CNN layer and then take the max-over-time pooling operation [4] to capture the most important features for each feature map. The vector output from the max-over-time pooling layer is the character-level representation of the word, called *Character Representation*. This representation is then concatenated with the word embedding as the input to word-level CNNs.

Highway Network. In our experiments, simply stacking multiplayer word-level CNNs makes the performance worse. Instead, we implement a highway network [12] after the CNN layer to keep valuable features across word-level CNNs. Highway layer allows part of s_i to be carried unchanged to the output while the reset to go through convolutional transformations.

BLSTM. In the next step, the output of highway network will be obtained for the input of a Long Short-term Memory Network (LSTM) [13], which was proposed to address this issue of learning long-term dependencies by maintains three multiplicative gates which control the information to forget and to pass on to the next step. In sequence tagging task, we apply a Bidirectional LSTM (BLSTM) to make use of past (left) and future (right) information by present

each sequence forwards and backwards to two separate hidden states. The two hidden states are then concatenated to form the final output.

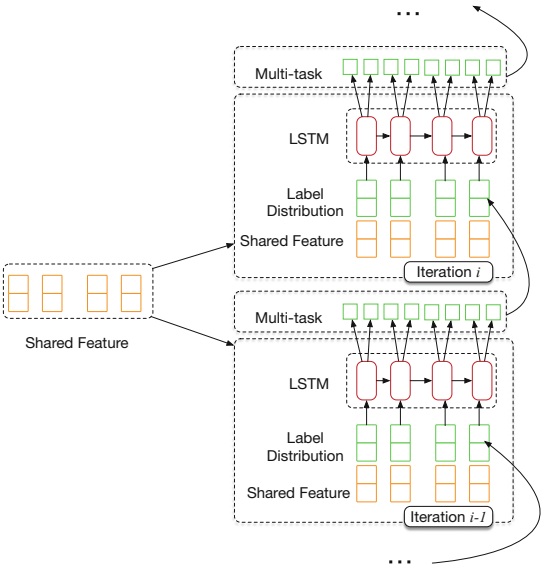

Fig. 1. MTL with recurrent iterative framework.

2.3 The Final Model

Our final MTL architecture is illustrated in Fig. 1. When iterative training begins, we initialize prediction probabilities of all tasks with zero vectors, as input of BLSTM for a unified form. It is worth noting that parameters in each iteration are not shared. Following previous work [4], our task-specific hidden layer $h^{(i)}$ only includes task-specific fully connected output layers.

3 Experiment

3.1 Experimental Setup

We test our model on three NLP sequence tagging tasks: POS tagging on Penn TreeBank (PTB), chunking on CoNLL 2000 and named entity recognition (NER) on CoNLL 2003.

POS assign a unique tag to each word, which indicate its syntactic role, such as noun, verb and so on. Chunking, or shallow parsing, assigns each word with its phrase type. NER labels each word into other or four categories: Person, Location, Organization, or Miscellaneous. We use the BIOES tagging scheme for chunking and NER tasks. For the preprocessing, we follows previous work [5].

In the training of model, we fine turn the parameter on the validation dataset. For the chunking task on CoNLL 2000 data, we split 10% training data for validation because the benchmark only provide training data. Since the test data for chunking in the CoNLL 2000 shared task is part of the validation data in the standard POS, we simply remove part of repeat samples from validation data of POS task. The size of training data we used in our multi-task learning is smaller than the standard task.

All the experimental result reported in our paper is with p-value less than 0.01 by t-test. For the metrics in evaluation, we follow the standard metrics on three benchmark respectively. In detail, for the POS tagging task, we use **Accuracy** to evaluate the performance. For the chunking and NER tasks, we evaluate the result by **F1-measure**.

3.2 Experimental Results

In this section, we first explore the effectiveness of recurrent iteration mechanism on several experiments. Then we report the results of our model on the benchmark and compare to the previously-reported state-of-art results.

Table 1. The impact of recurrent iteration.

Length	POS	Chunking	NER
T=0	97.54	95.31	90.52
T=1	**97.58**	95.47	91.16
T=2	97.56	95.54	**91.31**
T=3	97.56	**95.58**	91.30
T=4	97.55	**95.58**	91.26
T=5	97.55	95.57	91.24

Impact of Iterative Training. In this section, we first explore the effectiveness of our proposed iterative training procedure. Then we report the results of our model on the benchmarking datasets in comparison to previously reported state-of-the-art results.

Table1 shows how the results with the increasing number of iterations for the iterative training of our proposed base sequence tagging model. When iterative training just once (T=0), the model is common form MTL model as illustrate in Fig. 1. It can be observed that iterative training has little impact on POS tagging, but it improves the performance of both chunking and NER significantly, especially in the first two iteration. For computational efficiency, we set the number of iteration to 3 in the later experiment.

To understand the effect of the predicted tag sequences from one task on the others, we drop tag prediction results from one task at a time during the iterative

Table 2. The effect of excluding the predicted tag sequences from one task at a time during iterative training.

Models	POS	Chunking	NER
Full	**97.56**	**95.58**	**91.30**
Without POS	97.52	95.41	90.70
Without Chunking	97.53	95.51	90.93
Without NER	97.52	95.52	90.75

training process (i.e. the input of BLSTM at each iteration only includes the prediction results of two tasks and $h^{(shared)}$). From the results presented in Table 2, we find that POS tags have more noticeable influence on NER and chunking. Chunking tags also have some impact on NER, but do not influence much on POS tagging. NER tags do not contribute much to POS tagging or chunking. We also notice that NER tag sequences from previous training iteration are helpful for tagging prediction in the next iteration.

Comparison with Existing Systems. We first compare our method with the previous multi-task learning method [4], which used CNNs as base model, in Table 3. It can be observed that with our proposed base sequence tagging model and the iterative training process, our approach outperforms the method in [4] significantly on NER, noticeably on chunking and slightly on POS tagging.

Table 3. Compare with existing multi-task learning method.

Models	POS	Chunking	NER
Collobert et al., 2011	97.22	94.10	88.62
Our approach	**97.56**	**95.58**	**91.30**

We also compare our results with methods developed specifically for POS tagging, chunking and NER in Table 5, respectively. For POS tagging, the best performing model is the one proposed in [14] where an accuracy of 97.78% was achieved by using character-level and word-level BLSTM model. The word embeddings used in their model are trained by themselves and are not publicly available. [1] achieved an accuracy of 97.55% by using CNN as the character-level model and BLSTM-CRF as the word-level model. Our model slightly outperforms [1] demonstrating the effectiveness of using CNN-Highway for modeling word-level local features. Overall, our model outperforms all the other systems apart from [14], including the ones using handcrafted features. For chunking methods, [15] won the CoNLL2000 challenge with an F1-score of 93.48% by using SVMs. The previous state-of-the-art F1-score of 95.23% was reported in [16] by using a voting classifier scheme with carefully handcrafted features.

Table 4. Compare with existing POS tagging methods, Chunking Methods and NER Methods. (The methods with handcraft features have been marked with †.)

Methods	POS Tagging	Chunking	NER
Sha adn Pereira, 2003	-	94.30	-
Shen et al., 2005 †	-	95.23	-
Collobert et al., 2011	97.29	94.32	89.59
Lample et al., 2015	**97.78**	-	90.94
Ma et al., 2016	97.55	-	91.21
Huang et al., 2015	97.55	-	90.10
Our approach	97.56	**95.58**	**91.30**

Our multi-task learning model outperforms all the existing methods on chunking, partly attributing to the additional training data from POS tagging and NER tasks and also common features extracted by the shared lower layer.

For NER methods, with the same data pre-processing as ours, [5] obtained an F1-measure of 90.94% by using the character-level BLSTM and word-level BLSTM-CRF model. [17] achieved an F1-score of 91.20% using the joint NER and entity linking model with heavily handcrafted features. With our proposed iterative training process, our model improves upon [17] by 0.10% in F1-score. To the best of our knowledge, the previous best result of 91.21% was reported in [1] with a BLSTM-CNNs-CRF model. Our model further boosts the performance by 0.09% and achieves the state-of-art performance.

3.3 Case Study

To validate our assumption of iterative training, we choose an example in test set of chunking task to show the iterative tagging results in Table 5:

Ex.1 *The Canadian Wheat Board reported six ships loading*

In Table 6, the top row illustrates the ground truth of POS and Chunk labels, and the next three rows show the iterative tagging results in validation phase. We can obverse that, without iterative training, our model is not able to predict the right chunking label of word "*loading*". However, the POS label of "*loading*" is correctly predicted and it is used to help correct the label of chunking when T=1. In second iteration, by leveraging the sequential property of text, our model also replaces the chunking label of "*ships*" from "*E-NP*" to "*I-NP*". We also find that, with increasing number of iterations, the tagging results of our model just remain stable.

4 Related Work

In recently years, a number of neural architectures have been proposed for sequence tagging. [3] proposed a BLSTM-CRF model for sequence tagging

Table 5. Tagging results of different iterations

	Tag	Reported	Six	Ships	Loading
Ground	POS	VBD	CD	NNS	NN
	Chunk	S-VP	B-NP	I-NP	E-NP
T=0	POS	VBD	CD	NNS	NN
	Chunk	S-VP	B-NP	E-NP	S-VP
T=1	POS	VBD	CD	NNS	NN
	Chunk	S-VP	B-NP	E-NP	E-NP
T=2	POS	VBD	CD	NNS	NN
	Chunk	S-VP	B-NP	I-NP	E-NP
T=3	POS	VBD	CD	NNS	NN
	Chunk	S-VP	B-NP	I-NP	E-NP
T=4	POS	VBD	CD	NNS	NN
	Chunk	S-VP	B-NP	I-NP	E-NP

tasks. The experimental results on POS tagging, chunking and NER tasks have obtained impressive results. However, the handcrafted features they used make the model less flexible. [6] proposed a hybrid of BLSTM and CNNs for NER based on character-type, capitalization and lexicon features. To reduce handcrafted features, [2] proposed CharWNN, which stacks a convolutional layer to capture word morphology and shape features. It obtains the state-of-the-art accuracy on English POS tagging. More recently, [5] proposed a BLSTM-CRF model for NER based on character-level and word-level information. [1] proposed a LSTM-CNNs-CRF model for POS tagging and NER, which included character-level CNN layers and word-level LSTM-CRF layers. Both [1,5] utilized neural network that learns character-level representation of words instead of using handcrafted features. All models mentioned above are trained on a single task and can not leveraged datasets of related tasks.

More recently, multi-task learning has been applied to various sequence tagging tasks, including name error recognition [8], POS [18]. Most of these models use shared representation constructed by lower layers and task-specific representation constructed by upper layers. [4] utilized multi-task learning for POS, Chunking, and NER joint tagging. However, this work did not consider the predicted tagging sequences from each individual task and the base model is a simple CNNs, which limited the performance. [19] jointly trained co-reference resolution, entity linking, and NER using a single CRF model with added cross-task interaction factors, which also could capture interaction of tags between related tasks. However, jointly training a CRF model requires all tasks having fully labeled training data and the model can not leverage exiting partial labeled data for training.

5 Conclusion

In this paper, we have presented a new multi-task learning network architecture for sequence tagging. In this method, CNNs was used to model both character-level and word-level representations, and BLSTM was used to capture long range dependencies. We also utilize a Highway network to further improve the model performance. A main contribution of our proposed framework is that it can leverage the predicted tagging sequences between related tasks through the iterative training procedure. Our model achieves the state-of-the-art performance on chunking and NER, and performs comparably to the best performing model on POS tagging, without using any external knowledge or handcrafted features. Since our model does not require any domain- or task-specific knowledge, it can be applied to other sequence tagging tasks, which will be explored in our future work.

Acknowledgement. This work was supported by the National Natural Science Foundation of China U1636103, 61632011, Shenzhen Foundational Research Funding 20170307150024907, Key Technologies Research and Development Program of Shenzhen JSGG20170817140856618.

References

1. Ma, X., Hovy, E.H.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: ACL, pp. 1064–1074 (2016)
2. dos Santos, C.N., Zadrozny, B.: Learning character-level representations for part-of-speech tagging. In: ICML, pp. 1818–1826 (2014)
3. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991 (2015)
4. Collobert, R., et al.: Natural language processing (almost) from scratch. JMLR **12**, 2493–2537 (2011)
5. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: NAACL, pp. 260–270 (2016)
6. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNS. TACL **4**, 357–370 (2016)
7. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
8. Cheng, H., Fang, H., Ostendorf, M.: Open-domain name error detection using a multi-task RNN. In: EMNLP, pp. 737–746 (2015)
9. Søgaard, A., Goldberg, Y.: Deep multi-task learning with low level tasks supervised at lower layers. In: ACL(2), pp. 231–235 (2016)
10. Alonso, H.M., Plank, B.: When is multitask learning effective? Semantic sequence predictionunder varying data conditions. In: EACL, pp. 44–53 (2017)
11. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: AAAI, pp. 2741–2749 (2016)
12. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: NIPS, pp. 2377–2385 (2015)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

14. Ling, W., Dyer, C., Black, A.W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., Luís, T.: Finding function in form: compositional character models for open vocabulary word representation. In: EMNLP, pp. 1520–1530 (2015)
15. Kudoh, T., Matsumoto, Y.: Use of support vector learning for chunk identification. In: CoNLL, pp. 142–144 (2000)
16. Shen, H., Sarkar, A.: Voting between multiple data representations for text chunking. In: Kégl, B., Lapalme, G. (eds.) AI 2005. LNCS (LNAI), vol. 3501, pp. 389–400. Springer, Heidelberg (2005). https://doi.org/10.1007/11424918_40
17. Luo, G., Huang, X., Lin, C.-Y., Nie, Z.: Joint entity recognition and disambiguation. In: EMNLP, pp. 879–888 (2015)
18. Plank, B., Søgaard, A., Goldberg, Y.: Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In: ACL(2), pp. 412–418 (2016)
19. Durrett, G., Klein, D.: A joint model for entity analysis: coreference, typing, and linking. TACL **2**, 477–490 (2014)