NITE: A Neural Inductive Teaching Framework for Domain-Specific NER

Siliang Tang, Ning Zhang, Jinjian Zhang, Fei Wu and **Yueting Zhuang**
College of Computer Science, Zhejiang University, China
{siliang, aning, jibjianzhang, wufei, yzhuang}@zju.edu.cn

Abstract

In domain-specific NER, due to insufficient labeled training data, deep models usually fail to behave normally. In this paper, we proposed a novel Neural Inductive TEaching framework (NITE) to transfer knowledge from existing domain-specific NER models into an arbitrary deep neural network in a teacher-student training manner. NITE is a general framework that builds upon transfer learning and multiple instance learning, which collaboratively not only transfers knowledge to a deep student network but also reduces the noise from teachers. NITE can help deep learning methods to effectively utilize existing resources (i.e., models, labeled and unlabeled data) in a small domain. The experiment resulted on Disease NER proved that without using any labeled data, NITE can significantly boost the performance of a CNN-bidirectional LSTM-CRF NER neural network nearly over 30% in terms of F1-score.

1 Introduction

Domain-specific Named Entity Recognition (DNER), which aims to identify domain specific entity mentions and their categories, plays an important role in domain document classification, retrieval and content analysis. It is also a foundation for further level of complex information extraction tasks, serves as cornerstone in the knowledge computing process of transforming data into machine readable knowledge (Zhuang et al., 2017). Domain-specific NER is a challenging problem. For example, in biomedical domain, the number of unseen biomedical entity mentions (such as disease names, chemical names), their abbreviations or acronyms, as well as multiple names of the same entity is growing fast with the rapid increase of biomedical literatures and clinical records. However, the performance of a learning based NER system relies heavily on data annotation, which is quite expensive. The situation is even worse in domain-specific NER systems, since their data annotation requires the engage of domain experts. Therefore, in many special domains, only trained models or APIs are available, while their training data are private and inaccessible. On the other hand, due to insufficient labeled training data, deep models usually fail to behave normally in such domain, and state-of-the-art methods in these domains are usually dominated by rule based deductive methods or shallow model with hand-crafted features. However, the way of pre-defining useful domain specific hand-crafted features or rules are usually unavailable to the public.

In this paper, we proposed a novel Neural Inductive TEaching framework (NITE) to transfer knowledge from existing models into an arbitrary deep neural network. The idea of NITE is mainly borrowed from Transfer learning (Pan and Yang, 2010) where previously learned knowledge can aid current situation and solve problems with better solutions. In NITE, existing NER models behave like inefficient teachers to teach a deep neural network (we called student network) to identify named entities by giving it concrete examples. The knowledge transferred from these models is their posterior distributions on unlabeled data. These teachers are inefficient because they transfer not only useful information, but also errors to the student. The inputs of student network can be twofold, one is a small proportion from human labeled ground truth data (optional, like text book), and another is a large proportion from teachers, which is always noisy and less trustable.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2635–2640
Copenhagen, Denmark, September 7–11, 2017.

In such case, a student is overwhelmed and often inferior to the teachers, therefore in NITE, we introduced Multiple Instance Learning (MIL) trick (Dietterich et al., 1997; Babenko, 2008) to reduce the input noise during the model training.

In summary, NITE is a general framework that can help deep learning methods to make the best use of existing resources (i.e., models, labeled and unlabeled data). The experiment results on Disease NER (DNER) proved that without using any labeled data, NITE can significantly boost the performance of a CNN-bidirectional LSTM-CRF NER neural network (Ma and Hovy, 2016), which trained on NCBI training dataset nearly over 30% in terms of F1-score. It also outperformed the teacher model, which proved the correctness of our hypothesis.

2 Neural Inductive Teaching Framework

In this section we will define our NITE framework step by step, and apply it to Disease NER.

2.1 Inductive Teaching

Inductive teaching means teaching student by examples, our inductive teaching method builds upon teacher-student models (Ba and Caruana, 2014) and knowledge distillation (Hinton et al., 2015). The main idea of our method is to transfer discriminative knowledge from well-trained existing models (teachers) to a new and more capable model (student). The student learns by imitating the teachers' behaviors, and the teaching process can be defined as follows:

Let $x = \{w_1, w_2, \ldots, w_{|x|}\}$ be an input sentence of $|x|$ words, where w_k is the kth word in x. If l_k is the corresponding 3-dimensional one hot IOB (In-Out-Begin) vector for w_k, then the NER labeling sequence of x can be defined as $y = \{l_1, l_2, \ldots, l_{|x|}\}$.

For a given sentence x_i, we further define the posterior distribution of a teacher as $y_i^{f_t} = f_t(y_i|x_i)$, while the posterior distribution of a student network can be defined as $y_i^{f_s} = f_s(y_i|x_i;\theta)$, where θ is the parameters of the student network. During training, we measure the similarity between y^{f_t} and y^{f_s} with KL-divergence, and minimize their difference. Therefore, for a given x_i, we optimize:

$$\min_{\theta} \sum_{i=1}^{n} D_{KL}(f_t(y_i|x_i)||f_s(y_i|x_i;\theta)) \quad (1)$$

, where $D_{KL}(P||Q) = \sum_j P_j \log \frac{P_j}{Q_j}$ is the KL-divergence. This equation can be optimized through stochastic gradient descent over shuffled mini-batches with the Adadelta (Zeiler, 2012) update rule.

2.2 Multiple Instance Learning

Multiple Instance Learning is an effective training method that can help to train a supervised model to alleviate the wrong label problem (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012). Instead of predicting labels for each individual training sample, the objective of MIL is to predict the labels (positive or negative) of the unseen bags, where each bag contains a fixed number of instances (samples). The standard MIL assumption assumes that a bag is positively labeled if at least one instance in a bag is positive, and is negatively labeled if all instances in a bag are negative. MIL is generally used in training a binary classifier, to apply MIL in NITE, we redefine the label of a bag as the quality (correctness) of its containing samples. Thus, in NITE, a bag is positively labeled if at least one instance in it is labeled correctly. Furthermore, it is inappropriate to evaluate the correctness of IOB label (i.e., l_k) of each word (i.e., w_k), since the IOB sequence y_i of a sentence x_i is generated dependently. Therefore, we choose sentence x_i as our MIL instance, and the correctness of x_i is evaluated by the likelihood probability of all words with correct BIO tags. In general, our MIL can be formally defined as follows:

Randomly allocate training samples in a mini-batch $\mathcal{B}$ into M bags, i.e., $\mathcal{B} = \{B_1, B_2, \ldots, B_M\}$ with their corresponding labels $\{z_1, z_2, \ldots, z_M\}$, where $z_m \in \{-1, 1\}$. For bag B_m, it contains K instances, i.e., $B_m = \{x_1, x_2, \ldots, x_K\}$, where x_i is a sentence with its posterior evaluation $y_i^{f_s}$.

During the training, given a bag B_m, if $z_m = 1$, which means B_m is a positive bag. In order to reduce the noise, our MIL learner will select the most correct instance $y_{i*}^{f_s}$, which has the maximum likelihood among all other instances (i.e., sentence) in the bag B_m. That is $P(z_m = 1|B_m) = P(y_{i*}^{f_s}) = \arg\max_i \{P(y_i^{f_s}|x_i)\}$, where $1 \leq i \leq K, x_i \in B_m$. If $z_m = -1$, which means B_m is a negative bag, in order to better detect such negative bags, our MIL learner should select the most violated instance for learning, which is also the instance with maximum likelihood. Thus, the bag label z (which indicates the sentence is labeled

correctly or incorrectly) is actually integrated out, since no matter what the value z is, MIL in NITE will always select the instance with the highest likelihood probability. Finally the MIL in NITE can be summarized as:

$$P(z_m|B_m) = P(y_{i*}^{f_s}) = \arg\max_{i=1:K}\{P(y_i^{f_s}|x_i)\} \quad (2)$$

In summary, MIL in NITE can be regarded as a mechanism for posterior selection, or regularization on posterior distribution of a student network. Therefore, MIL only affects the model training, and it will not affect the testing process.

2.3 Teacher Model & Student Network

Theoretically, the teacher model of NITE can be any existing well-trained model, while the student network can be an arbitrary deep neural network. In this paper, we focus on domain-specific NER, and more specifically on Disease NER, which is a small but typical domain that is suffering from insufficient labeled training data.

There are many existing DNER systems, and the most well-known systems are BANNER (Leaman et al., 2008), and DNorm (Leaman et al., 2013). BANNER is an open-source biomedical NER system implemented using conditional random fields (CRFs) (Lafferty et al., 2001). While, DNorm uses supervised semantic indexing, is trained with pairwise learning to rank, to score the mentions returned by BANNER. Therefore, DNorm can be regarded as an extension of BANNER, and the whole system depends on handcrafted features such as word spelling features and orthographic features. DNorm is the state-of-the-art DNER system, and therefore we adopt DNorm as our teacher model.

For the student network, we are looking for state-of-the-art solutions in general NER. There are many studies on applying complex deep learning models on general NER or other sequence labeling tasks. Without any feature engineering trick, deep models have achieved comparable or better performances than many other traditional methods. More recently, Ma and Hovy (2016) proposed a method that concatenated CNN, bidirectional LSTM, and CRF successively to form an end to end deep NER model (CLC for short). CLC achieved state-of-the-art performance in general NER, and therefore we take the CLC as our student network, Fig. 1 shows the overall architecture of our student network.

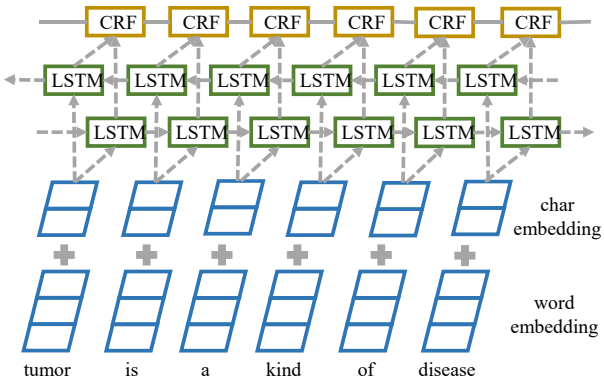

Figure 1: the Flowchart of the Student Network

As shown in Fig. 1, the character-level embeddings are generated by CNN layers, then are concatenated with pre-trained word embeddings, and finally fed into the bidirectional LSTM layer. The bidirectional LSTM is efficient to capture syntactic and semantic information both preceding and following simultaneously. Its output vectors are fed into the CRFs layer for IOB sequence labeling. It uses maximum conditional likelihood estimation to choose parameters during the finally CRFs training process, and its likelihood can be given as follows:

$$P(y_i^{f_s}|x_i) = \arg\max_{y\in\mathcal{Y}(x_i)} P(y|x_i) \quad (3)$$

, where $\mathcal{Y}(x_i)$ denotes the set of possible label sequences for x_i. Eq. 3 can be solved efficiently by adopting the Viterbi algorithm.

Fig. 2 shows the whole NITE-NER training process. For each training iteration, training samples in a mini-batch are randomly allocated into M bags, and then fed into the student network f_s. For bag B_m, the student network will generate posterior evaluation $y_i^{f_s}$ for each input instance $x_i \in B_m$ respectively. Then the MIL module will select the best sample $y_{i*}^{f_t}$ from all K instances according to Eq. 3 and 2. Finally, NITE will retrieve posterior evaluation $y_{i*}^{f_t}$ from the teacher, and update θ based on Eq. 1.

3 Experiments

In this section we designed several experiments to testify our hypothesis of inductive teaching as well as evaluate our NITE framework.

3.1 Training Corpus

Although NITE is a supervised learning framework, the discriminative knowledge of student net-

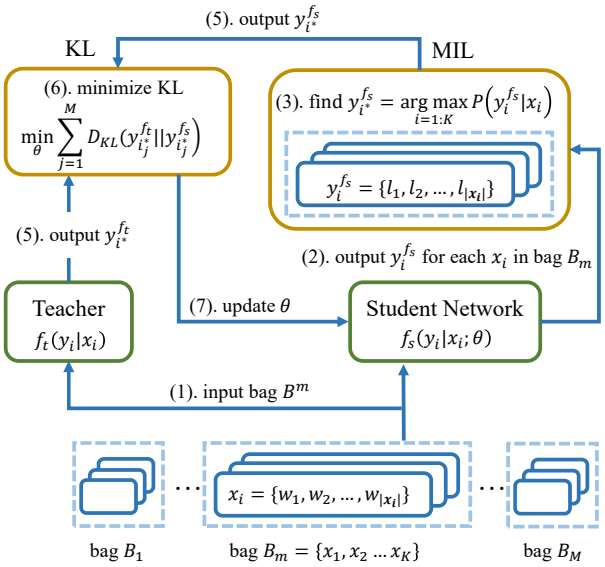

Figure 2: The training process of NITE-NER.

work is learned indirectly from the teacher models, therefore NITE can be trained without any labeled data.

To evaluate the efficiency of the NITE framework, we trained two DNER models on NCBI disease corpus (Doğan et al., 2014; Islamaj Dogan and Lu, 2012). One is the well-known DNorm model, which is the state-of-the-art method in disease NER. Another one is the bi-directional LSTM-CNN-CRF NER neural network i.e., CLC (Ma and Hovy, 2016), which has the state-of-the-art performance in general NER task. The CLC architecture also serves as our student network.

The NCBI disease corpus is a widely used data corpus with disease name and related concept annotations in biomedical research field. The corpus is an extension of the AZDC corpus (Leaman et al., 2009) which was annotated only with disease mentions. The detailed characteristics of the NCBI disease corpus as well as how we partition the data are shown in Table 1.

3.2 Experiment Setup

The experiment's setup is as follows:

Our NITE-DNER is trained without any labeled data, we randomly sampled 2,000 unlabeled abstracts of biomedical literature from PubMed as our training data. The DNorm model is served as the teacher model in the NITE framework.

In student network, we initialized character embeddings with uniform samples from $[-\sqrt{\frac{3}{d}}, +\sqrt{\frac{3}{d}}]$, where we set the dimension $d = 30$. We use 30 filters with window length 3 in CNN and 200 hidden states in bi-directional LSTM. In training procedure we set initial learning rate $\eta_0 = 0.015$ with decay rate $\rho = 0.05$, the learning rate is updated as $\eta_t = \eta_0/(1.0 + \rho n)$, where n is the number of epochs. We use a fixed dropout rate 0.5 at CNN and both input and output vectors of bi-directional LSTM to mitigate overfitting. For MIL we set the bag size $K = 5$ with mini-batch size 30. We implemented neural networks on a GeForce GTX 1080 using Theano.

NCBI	Train	Validate	Test
# of documents	593	100	100
# of sentences	5661	791	961
# of disease	5148	791	961
Specific Disease	2959	409	556
Disease Class	781	127	121
Modifier	1292	218	264
Composite Mention	116	37	20

Table 1: The description of the NCBI corpus as training, validating and testing sets for the recognition of disease named entity

3.3 Results and Discussion

We evaluated all three DNER methods on the NCBI test set in terms of precision, recall and F1-score. All the measurements are based on exact location of extracted disease mentions in the given test sentences.

Method	CLC-DNER	DNorm	NITE
Labels	NCBI	NCBI	-
Remark	Student only	Teacher only	S+T+MIL
Precision	79.20	80.50	85.40
Recall	51.73	75.70	75.07
F1-score	62.58	78.06	**79.91**

Table 2: Performance comparisons.

The experiment results are presented in Table 2. As shown in Table 2, although the complex CLC network is the state-of-the-art method in general NER, it behaves poorly in domain-specific NER task due to insufficient labeled training data. However, with the help of our NITE framework, its performance is significantly boosted, and reached the comparable level of DNorm. This proved that knowledge transfer in NITE is efficient and important in training a deep model of domain-specific NER.

3.4 Conclusion

In this paper, we proposed a general framework, NITE, and demonstrated its efficiency in transferring DNER knowledge into an end to end deep NER model. Although we only proposed a solution for DNER, it could be easily applied to other domain-specific NER problems (e.g., chemical, gene, and protein) or even applications other than NER. The experiment results suggested that NITE can be very helpful on training a deep model when other resources are available. For future work, a NITE architecture with more than one teacher could be considered. Moreover, as mentioned in (Zhou et al., 2017), crowd knowledge can be used to reshape deep learning features. Our framework can also incorporate crowd knowledge easily, in which the teachers can be human crowds, and then the NITE can employs active learning (Olsson, 2009) or lifelong machine learning (Chen and Liu, 2016) to progressively polishing the student model.

Acknowledgments

This work was supported in part by the 973 program (No. 2015CB352302), NSFC (No. 61625107, U1611461, 61402401), Chinese Knowledge Center of Engineering Science and Technology (CKCEST), Qianjiang Talents Program of Zhejiang Province 2015 and Key program of Zhejiang Province (2015C01027).

References

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662.

Boris Babenko. 2008. Multiple instance learning: algorithms and applications. *View Article PubMed/NCBI Google Scholar*.

Zhiyuan Chen and Bing Liu. 2016. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Rezarta Islamaj Dogan and Zhiyong Lu. 2012. *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, chapter An improved corpus of disease mentions in PubMed citations. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, page btt474.

Robert Leaman, Graciela Gonzalez, et al. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific symposium on biocomputing*, volume 13, pages 652–663.

Robert Leaman, Christopher Miller, and G Gonzalez. 2009. Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. In *Proceedings of the 2009 Symposium on Languages in Biology and Medicine*, volume 82.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Le-kui Zhou, Si-liang Tang, Jun Xiao, Fei Wu, and Yue-ting Zhuang. 2017. Disambiguating named entitieswith deep supervised learning via crowd labels. *Frontiers of Information Technology & Electronic Engineering*, 18(1):97–106.

Yue-ting Zhuang, Fei Wu, Chun Chen, and Yun-he Pan. 2017. Challenges and opportunities: from big data to knowledge inai 2.0. *Frontiers of Information Technology & Electronic Engineering*, 18(1):3–14.