Employing Auto-annotated Data for Person Name Recognition in Judgment Documents

Limin Wang, Qian Yan, Shoushan Li(✉), and Guodong Zhou

Natural Language Processing Lab, School of Computer Science and Technology,
Soochow University, Suzhou, China
{lmwang, qyan}@stu.suda.edu.cn, {lishoushan,
gdzhou}@suda.edu.cn

Abstract. In the last decades, named entity recognition has been extensively studied with various supervised learning approaches depend on massive labeled data. In this paper, we focus on person name recognition in judgment documents. Owing to the lack of human-annotated data, we propose a joint learning approach, namely Aux-LSTM, to use a large scale of auto-annotated data to help human-annotated data (in a small size) for person name recognition. Specifically, our approach first develops an auxiliary Long Short-Term Memory (LSTM) representation by training the auto-annotated data and then leverages the auxiliary LSTM representation to boost the performance of classifier trained on the human-annotated data. Empirical studies demonstrate the effectiveness of our proposed approach to person name recognition in judgment documents with both human-annotated and auto-annotated data.

Keywords: Named entity recognition · Auto-annotated data · LSTM

1 Introduction

Named entity recognition (NER) is a natural language processing (NLP) task and plays a key role in many real applications, such as relation extraction [1], entity linking [2], and machine translation [3]. Named entity recognition was first presented as a subtask on MUC-6 [4], which aims to find organizations, persons, locations, temporal expressions and number expressions in text. The proportion of Chinese names in the entities is large, according to statistics, in the "People's Daily" in January 1998 corpus (2,305,896 words), specifically, the average per 100 words contains 1.192 unlisted words (excluding time words and quantifiers), of which 48.6% of the entities are Chinese names [5]. In addition to the complex semantics of Chinese, the Chinese name has a great arbitrariness, so the identification of the Chinese name is one of the main and difficult tasks in named entity recognition.

In the paper, we focus on the person name recognition in judgment documents. The ratio of person name in judgment documents is very big, including not only plaintiffs, defendants, entrusted agents, but also other unrelated names, such as outsider, eyewitness, jurors, clerk and so on. For instance, Fig. 1 shows an example of a judgment document where person names exist. However, in most scenarios, there is insufficient

M. Sun et al. (Eds.): CCL 2017 and NLP-NABD 2017, LNAI 10565, pp. 13–23, 2017.
https://doi.org/10.1007/978-3-319-69005-6_2

annotated corpus data for person name recognition in judgment document and to obtain such corpus data is extremely costly and time-consuming.

```
民事 裁定书
（2016）川
原告 <ENAMEX TYPE="PERSON">阿衣子</ENAMEX>
被告 <ENAMEX TYPE="PERSON">艾现英</ENAMEX>
……
陪审员<ENAMEX TYPE="PERSON">胡士戎</ENAMEX>
书记员<ENAMEX TYPE="PERSON"> 丁丁</ENAMEX>
(English Translation:
Civil Judgment
(2016) Chuan
  Plaintiff <ENAMEX TYPE="PERSON"> Yizi A</ENAMEX>
  Defendant <ENAMEX TYPE="PERSON">Xianyin Ai</ENAMEX>
  ……
  Jurors <ENAMEX TYPE="PERSON">Shirong Hu</ENAMEX>
  Clerk <ENAMEX TYPE="PERSON">Ding Ding</ENAMEX>
)
```

Fig. 1. An example of a judgment document with the person names annotated in the text

Fortunately, we find that the judgment documents are well-structured in some parts. For example, in Fig. 1, we can see that in the front part, the word “原告 (Plaintiff)” often follows a person name. Therefore, to tackle the difficulty of obtaining human-annotated data, we try to auto-annotate much judgment documents with some heuristic rules. Due to the large scale of existing judgment documents, it is easy to obtain many auto-annotated sentences with person names and these sentences could be used as training data for person name recognition.

E1:

*原告 <ENAMEX TYPE=“PERSON”> **阿衣子** </ENAMEX> 诉称, 被告 <ENAMEX TYPE=“PERSON”> **艾现英** </ENAMEX> 和她的邻居**高山**一起曾经带着案外人**方亮**出现在其出租屋。……*

(**English Translation:**

Plaintiff <ENAMEX TYPE=“PERSON”> Yizi A </ENAMEX> complained, defendant <ENAMEX TYPE=“PERSON”> Xianyin Ai </ENAMEX>, along with her neighbor GaoShan, had brought outsider FangLiang appearing in her rental. ……

)

One straightforward approach to using auto-annotated data in person name recognition is to merge them into the human-annotated data and use the merging data to train a new model. However, due to the automatic annotation, the data is noisy. That is to say, there still exist some person names are not annotated. For example, in E1, there are four person names in the sentence, but we can only annotate two person names via the auto-annotating strategy.

In this paper, we propose a novel approach to person name recognition by using auto-annotated data in judgment documents with a joint learning model. Our approach uses a small amount of human-annotated samples, together with a large amount of auto-annotated sentences containing person names. Instead of simply merging the human-annotated and auto-annotated samples, we propose a joint learning model, namely Aux-LSTM, to combine the two different resources. Specifically, we first separate the twin person name classification task using the human-annotated data and the auto-annotated data into a main task and an auxiliary task. Then, our joint learning model based on neural network develops an auxiliary representation from the auxiliary task of a shared Long Short-Term Memory (LSTM) layer and then integrates the auxiliary representation into the main task for joint learning. Empirical studies demonstrate that the proposed joint learning approach performs much better than using the merging method.

The remainder of this paper is organized as follows. Section 2 gives a brief overviews of related work on name recognition. Section 3 introduces data collection and annotation. Section 4 presents some basic LSTM approaches and our joint learning approach to name recognition. Section 5 evaluates the proposed approach. Finally, Sect. 6 gives the conclusion and future work.

2 Related Work

Although the study of Chinese named entities is still in the immature stage compared with the English named entity recognition. But there is a lot of research on Chinese named recognition. Depending on the method used, these methods can be broadly divided into three categories: rule method, statistical method and a combination of rules and statistics.

The rule method mainly uses two kinds of information: the name classification and the restrictive component of the surname: that is, when mark the name with the obvious character in the analysis process, the recognition process of the name is started and the relevant component, which limits the position of the name before and after.

In the last decades, named entity recognition has been extensively studied with various supervised shallow learning approaches, such as Hidden Markow Models (HMM) [6], sequential perceptron model [7], and Conditional Random Fields (CRF) [8]. Meanwhile, named entity recognition has been performed in various styles of text, such as news [6], biomedical text [9], clinical notes [10], and tweets [11].

An important line of previous studies on named entity recognition is to improve the recognition performance by exploiting extra data resources. One major kind of such researches is to exploit unlabeled data with various semi-supervised learning approaches, such as bootstrapping [12, 13], word clusters [14], and Latent Semantic Association (LSA) [15]. Another major kind of such researches is to exploit parallel corpora to perform bilingual NER [16, 17].

Recently, deep learning approaches with neural networks have been more and more popular for NER. Hammerton [18] applies a single-direction LSTM network to perform NER with a combination word embedding learning approach. Collobert [19] employs convolutional neural networks (CNN) to perform NER with a sequence of word embeddings. Subsequently, recent studies perform NER with some other neural networks, such as BLSTM [20], LSTM-CNNs [21], and LSTM-CRF [22].

3 Data Collection and Annotation

3.1 Human-annotated Data

The data is built by ourselves and it is from a kind of law documents named judgments. Choosing this special kind of document as our experimental data is mainly due to the fact that judgments always have an invariant structure and several domain-specific regulations could be found therein, which makes it a good choice to test the effectiveness of our approach. We obtain the Chinese judgments from the government public website (i.e., http://wenshu.court.gov.cn/). The judgments are organized in various categories of laws and we pick the Contract Law. In the category, we manually annotate 100 judgment documents according the annotation guideline in OntoNotes 5.0 [23]. Two annotators are asked to annotate the data. Due to the clear annotation guideline, the annotation agreement on name recognition is very high, reaching 99.8%.

3.2 Auto-annotated Data

Note that a Chinese judgment always has an invariant structure where plaintiffs and defendants are explicitly described in two lines in the front part. It is easy to capture some entities from two textual patterns, for example, "原告 NAME1, (Plaintiff NAME1,)" and "被告 NAME2, (Defendant NAME2,)" where "NAME1" or "NAME2" denotes a person name if the length is less than 4. Therefore, we first match the name through the rules in the front part of judgment instruments. Second, we only selected the sentences containing the person name as the auto-annotated samples from the entire judgment documents. In this way, we could quickly obtain more than 10,000 auto-annotated judgment documents.

4 Methodology

4.1 LSTM Model for Name Recognition

In this subsection, we propose the LSTM classification model. Figure 2 shows the framework overview of the LSTM model for name recognition.

Formally, the input of the LSTM classification model is a character's representation x_i, which consists of character unigram and bigram embeddings for representing the current character, i.e.,

$$x_i = v_{c_{i-1}} \oplus v_{c_i} \oplus v_{c_{i+1}} \oplus \ldots \oplus v_{c_{i+1},c_{i+2}} \quad (1)$$

Where $v_{c_i} \in R^d$ is a d-dimensional real-valued vector for representing the character unigram c_i and $v_{c_i,c_{i+1}} \in R^d$ is a d-dimensional real-valued vector for representing the character bigram c_i, c_{i+1}.

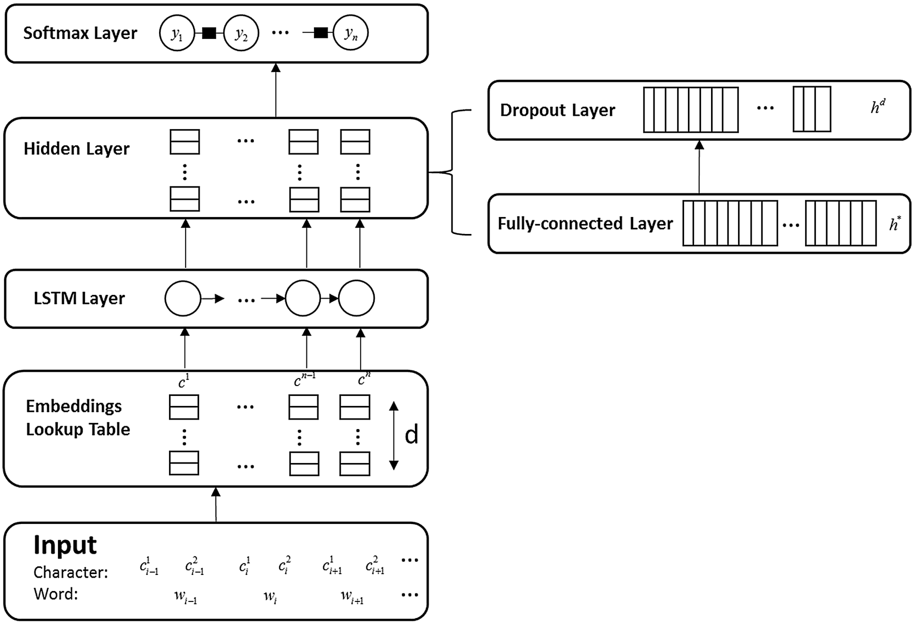

Fig. 2. The framework overview of the LSTM model for character-level NER

Through the LSTM unit, the input of a character is converted into a new representation h_i, i.e.,

$$h_i = LSTM(x_i) \tag{2}$$

Subsequently, the fully-connected layer accepts the output from the previous layer, weighting them and passing through a normally activation function as follows:

$$h_i^* = dense(h_i) = \phi(\theta^T h_i + b) \tag{3}$$

Where $\phi(x)$ is a non-linear activation function, employed “relu” in our model. h_i^* is the output from the fully-connected layer.

The dropout layer is applied to randomly omit feature detectors from network during training. It is used as hidden layer in our framework, i.e.,

$$h_i^d = h_i^* \cdot D(p^*) \tag{4}$$

Where D denotes the dropout operator, p^* denotes a tunable hyper parameter, and h_i^d denotes the output from the dropout layer.

The softmax output layer is used to get the prediction probabilities, i.e.,

$$P_i = softmax(W^d h_i^d + b^d) \tag{5}$$

Where P_i is the set of predicted probabilities of the word classification, W^d is the weight vector to be learned, and the b^d is the bias term. Specifically, P_i consists of the posterior probabilities of the current word belonging to each position tag, i.e.,

$$P_i = < p_{i,B-PER}, p_{i,I-PER}, p_{i,E-PER}, p_{i,O} > \tag{6}$$

4.2 Joint Learning for Person Name Recognition via Aux-LSTM

In the Fig. 3 delineates the overall architecture of our Aux-LSTM approach which contains a main task and an auxiliary task. In our study, we consider the person name recognition with the human-annotated data as the main task and the name recognition with auto-annotated data as the auxiliary task. The approach aims to enlist the auxiliary representation to assist in the performance of the main task. The main idea of our Aux-LSTM approach is that the auxiliary LSTM layer is shared by both the main and

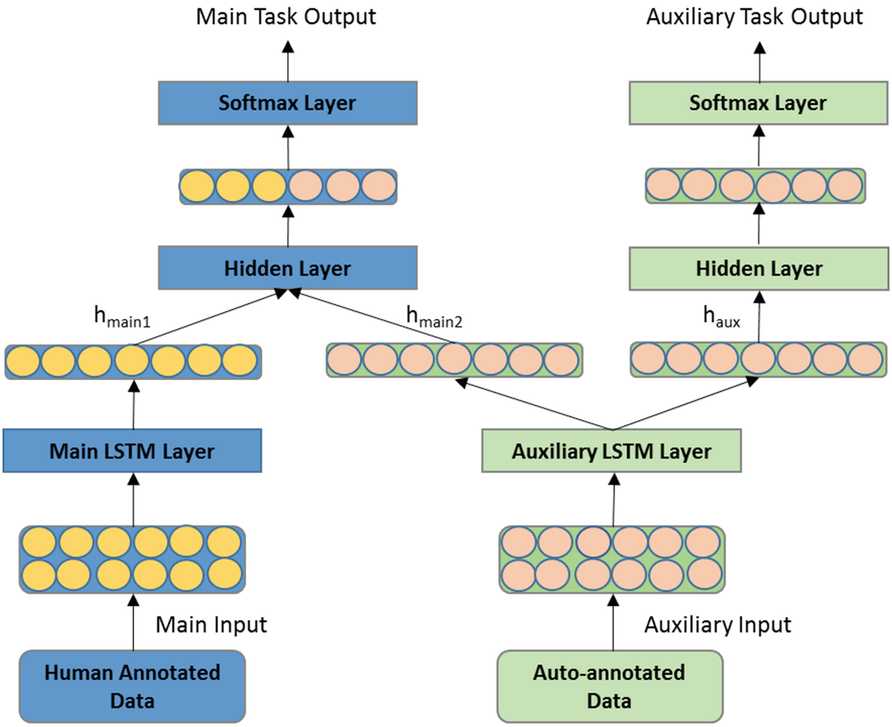

Fig. 3. Overall architecture of Aux-LSTM

auxiliary task so as to take advantage of information from both the annotated and auto-annotated data.

(1) **The Main Task:**

Formally, the representation of main task is generated from both the main LSTM layer and the auxiliary LSTM layer respectively:

$$h_{main1} = LSTM_{main}(T^{input}) \tag{7}$$

$$h_{main2} = LSTM_{aux}(T^{input}) \tag{8}$$

where h_{main1} represents the output of classification model via main LSTM layer and h_{main2} represents the output of classification model via auxiliary LSTM layer.

Then we concatenate the two representation as the input of the hidden layer in the main task:

$$h^{d}_{main} = dense_{main}(h_{main1} \oplus h_{main2}) \tag{9}$$

where h^{d}_{main} denotes the outputs of fully-connected layer in the main task, and $\oplus$ denotes the concatenate operator as a 'concat' mode.

(2) **The Auxiliary Task:**

The auxiliary classification representation is also generated by the auxiliary LSTM layer, which is a shared LSTM layer and is employed to bridge across the classification models. The shared LSTM layer encodes both the same input sequence with the same weights and the output h_{aux} is the representation for the classification model via shared LSTM model.

$$h_{aux} = LSTM_{aux}(T^{input}) \tag{10}$$

Then a fully-connected layer is utilized to obtain a feature vector for classification, which is the same as the hidden layer in the main task:

$$h^{d}_{aux} = dense_{aux}(h_{aux}) \tag{11}$$

Other layers such as softmax layer, as shown in Fig. 2, are the same as those which have been described in Sect. 4.1.

Finally, we define our joint cost function for Aux-LSTM as a weighted linear combination of the cost functions of both the main task and auxiliary task as follows:

$$loss_{Aux-LSTM} = \lambda(loss_{main}) + (1-\lambda)(loss_{aux}) \tag{12}$$

In the above equation, λ is the weight parameter, $loss_{main}$ and $loss_{aux}$ is the loss function of main task and auxiliary task respectively. We take 'adadelta' as the optimizing algorithm. All the matrix and vector parameters in neural network are initialized with uniform samples in $\left[-\sqrt{6/(r+c)}, \sqrt{6/(r+c)}\right]$, where r and c are the numbers of rows and columns in the matrices [24].

5 Experimentation

In this section, we have systematically evaluated our approach to person name recognition together with both human annotated and the auto-annotated data.

5.1 Experimental Settings

Data Setting: The data collection has been introduced in Sect. 3.1. In the main task, we randomly select 20 articles of human-annotated data as training data and another 50 articles of human-annotated as the test data. In the auxiliary task, we randomly select the number of training samples corresponding to the number of 5 times, 10 times, 20 times, 30 times and 40 times as the training data and the test data is the same as that in the main task.

Features and Embedding: We use the current character and its surrounding characters (window size is 2), together with the character bigrams as features. We use word2vec (http://word2vec.googlecode.com/) to pre-train character embeddings using the two data sets.

Basic Classification Algorithms: (1) Conditional Random Fields (CRFs), one popular supervised shallow learning algorithms, is implemented with the CRF++-0.53[1] and all the parameters are set as defaults. (2) LSTM, as the basic classification algorithm in our approach, is implemented with the tool Keras[2]. Table 1 shows the final hyper-parameters of the LSTM algorithm.

Table 1. Parameter settings in LSTM

Parameter description	Value
Dimension of the LSTM layer output	128
Dimension of the full-connected layer output	64
Size of the batch	32
Dropout probability	0.5
Epochs of iteration	20

Hyper-parameters: The hyper-parameter values in the LSTM and Aux-LSTM model are tuned according to performances in the development data.

[1] https://www.crf.it/IT.

[2] https://github.com/fchollet/keras.

Evaluation Measurement: The performance is evaluated using the standard precision (P), recall (R) and F-score.

5.2 Experimental Results

In this subsection, we compare different approaches to person name recognition with both human-annotated and auto-annotated data. The implemented approaches are illustrated as follows:

- **CRF:** It is a shallow-learning model which has been widely employed in name recognition, and it simply merges the human-annotated data and the auto-annotated samples together as the whole training data.
- **LSTM:** It is deep learning model which has been widely employed in the natural language processing community, and the training data is the same as that in CRF.
- **Aux-LSTM:** This is our approach which develops an auxiliary representation for joint learning. In this model, we consider two tasks: one is the name recognition with the human-annotated data, and the other is the name recognition with the auto-annotated data. The approach aims to leverage the extra information to boost the performance of name recognition. The parameter λ is set to be 0.5.

Table 2 shows the number of characters, sentences and person names in auto-annotated documents with different sizes. From this table, we can see that, there are a great number of person names that could be automatically recognized in judgment documents. When 1000 documents are auto-annotated, there are totally 79411 recognized person names, which make the auto-annotated data a big-size training data for person name recognition.

Table 2. The number of character, sentence and person name in different auto-annotated data

Number of auto-annotated documents	Number of characters	Number of sentences	Number of person names
100	173370	7128	8970
200	317845	13690	17286
400	606795	26134	29810
600	895745	39703	47578
800	1184695	51502	62742
1000	1473645	64817	79411

Table 3 shows the performance of different approaches to person name recognition when different size of human-annotated and auto-annotated data are employed. Specifically, the first line named “0” means using only human-annotated data and the second line “100” means using both human-annotated data and 100 auto-annotated judgment documents. From this table, we can see that,

- When no auto-annotated data is used, the LSTM model performs much better than CRF, mainly due to its better performance on Recall.

Table 3. Performance comparison of different approaches to name recognition

	CRF			LSTM			Aux-LSTM		
	P	R	F	P	R	F	P	R	F
0	94.4	41.9	58.1	77.3	60.0	67.58	— —	— —	— —
100	96.1	74.5	83.9	94.2	82.9	88.2	92.7	84.5	**88.4**
200	97.3	80.3	88.0	94.1	86.2	90.0	95.9	90.5	**93.1**
400	97.6	82.8	89.6	96.3	86.3	91.0	95.3	91.7	**93.5**
600	98.2	85.0	91.1	96.4	85.5	90.6	94.0	90.4	**92.2**
800	98.1	86.7	92.0	96.3	87.0	91.4	96.6	94.2	**95.3**
1000	97.7	87.4	92.3	97.5	86.4	91.6	95.5	91.4	**93.4**

- When a small size of auto-annotated data is used, the LSTM model generally performs better than CRF in terms of F1 score. But when the size of auto-annotated data becomes larger, the LSTM model performs a bit worse than CRF in terms of F1 score. No matter the LSTM or CRF model is used, using the auto-annotated data always improves the person name recognition performances with a large margin.
- When the auto-annotated data is used, our approach, i.e., Aux-LSTM, performs best among the three approaches. Especially, when the size of the auto-annotated data becomes larger, our approach performs much better than LSTM. This is possibly because our approach is more robust for adding noisy training data.

6 Conclusion

In this paper, we propose a novel approach to person name recognition with both human-annotated and auto-annotated data in judgment documents. Our approach leverages a small amount of human-annotated samples, together with a large amount of auto-annotated sentences containing person names. Instead of simply merging the human-annotated and auto-annotated samples, we propose a joint learning model, namely Aux-LSTM, to combine the two different resources. Specifically, we employ an auxiliary LSTM layer to develop the auxiliary representation for the main task of person name recognition. Empirical studies show that using the auto-annotated data is very effective to improve the performances of person name recognition in judgment documents no matter what approaches are used. Furthermore, our Aux-LSTM approach consistently outperforms using the simple merging strategy with CRF or LSTM models.

In our future work, we would like to improve the performance of person name recognition by exploring the more features. Moreover, we would like to apply our approach to name entity recognition on other types of entities, such as organizations and locations in judgment documents.

Acknowledgments. This research work has been partially supported by three NSFC grants, No. 61375073, No. 61672366 and No. 61331011.

References

1. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of EMNLP, pp. 724–731 (2005)
2. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population. In: Proceedings of COLING, pp. 277–285 (2010)
3. Babych, B., Hartley, A.: Improving machine translation quality with automatic named entity recognition. In: Proceedings of the 7th International EAMT Workshop, pp. 1–8 (2003)
4. Chinchor, N.: MUC7 Named Entity Task Definition (1997)
5. Ji, N.I., Kong, F., Zhu, Q., Peifeng, L.I.: Research on chinese name recognition base on trustworthiness. J. Chin. Inf. Process. **25**(3), 45–50 (2011)
6. Zhou, G., Su, J.: Named entity recognition using an Hmm-based chunk tagger. In: Proceedings of ACL, pp. 473–480 (2002)
7. Collins, M.: Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In: Proceedings of EMNLP, pp. 1–8 (2002)
8. Finkel, J.R., Grenager, T. Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of ACL, pp. 363–370 (2005)
9. Yoshida, K., Tsujii, J.: Reranking for biomedical named entity recognition. In: Proceedings of BioNLP, pp. 209–216 (2007)
10. Wang, Y.: Annotating and recognizing named entities in clinical notes. In: Proceedings of ACL-IJCNLP, pp. 18–26 (2009)
11. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: Proceedings of ACL, pp. 359–367 (2011)
12. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in NLP. In: Proceedings of ACL, pp. 264–271 (2007)
13. Brooke, J., Baldwin, T., Hammond, A.: Bootstrapped text-level named entity recognition for literature. In: Proceedings of ACL, Short Paper, pp. 344–350 (2016)
14. Brown, P.F., deSouza, P.V., Mercer, R.L., Della Pietra, V.J., Lai, J.C.: Classbased n-gram models of natural language. Comput. Linguist. **18**, 467–479 (1992)
15. Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., Su, Z.: Domain adaptation with latent semantic association for named entity recognition. In: Proceedings of NAACL, pp. 281–289 (2009)
16. Burkett, D., Petrov, S., Blitzer, J., Klein, D.: Learning better monolingual models with unannotated bilingual text. In: Proceedings of CONLL, pp. 46–54 (2010)
17. Che, W., Wang, M., Manning, C.D., Liu, T.: Named entity recognition with bilingual constraints. In: Proceedings of NAACL, pp. 52–62 (2013)
18. Hammerton, J.: Named entity recognition with long short-term memory. In: Proceedings of CONLL, pp. 172–175 (2003)
19. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
20. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CORR, abs/1508.01991 (2015)
21. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Trans. Assoc. Comput. Linguist. **4**, 357–370 (2016)
22. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT, pp. 260–270 (2016)
23. Hovy, E.H., Marcus, M.P., Palmer, M., Ramshaw, L.A., Weischedel, R.M.: Ontonotes: the 90% solution. In: Proceedings of NAACL-HLT, pp. 57–60 (2006)
24. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. J. Mach. Learn. Res. **9**, 249–256 (2010)

Springer

http://www.springer.com/978-3-319-69004-9

Chinese Computational Linguistics and Natural
Language Processing Based on Naturally Annotated Big
Data
16th China National Conference, CCL 2017, and 5th
International Symposium, NLP-NABD 2017, Nanjing,
China, October 13-15, 2017, Proceedings
Sun, M.; Wang, X.; Chang, B.; Xiong, D. (Eds.)
2017, XVIII, 482 p. 133 illus., Softcover
ISBN: 978-3-319-69004-9