M-CNER: A Corpus for Chinese Named Entity Recognition in Multi-Domains

Qi Lu, YaoSheng Yang, Zhenghua Li, Wenliang Chen, Min Zhang
School of Computer Science and Technology
Collaborative Innovation Center of Novel Software Technology and Industrialization
Soochow University, Suzhou 215006, China
luqibhf@qq.com, ysyang@stu.suda.edu.cn,
{ zhli13, wlchen, mingzhang}@suda.edu.cn

Abstract

In this paper, we present a new corpus for Chinese Named Entity Recognition (NER) from three domains : human-computer interaction, social media, and e-commerce. The annotation procedure is conducted in two rounds. In the first round, one sentence is annotated by more than one persons independently. In the second round, the experts discuss the sentences for which the annotators do not make agreements. Finally, we obtain a corpus which have five data sets in three domains. We further evaluate three popular models on the newly created data sets. The experimental results show that the system based on Bi-LSTM-CRF performs the best among the comparison systems on all the data sets. The corpus can be used for further studies in research community.

Keywords: Named Entity Recognition; Chinese Data Set; Information Extraction

1. Introduction

In recent years, there has been significant progress on the task of Named-Entity Recognition (NER) by using sequence labeling models in the settings of supervised learning, such as CRF and LSTM-CRF (Lafferty et al., 2001; Huang, Xu, and Kai 2015). NER is one of the most important natural language processing (NLP) tasks. Its performance highly affects further applications, such as relation extraction (Bunescu and Mooney, 2005), and question answering (Jurafsky and Martin, 2009).

As with the setting of supervised learning, building NER systems needs a massive amount of labeled training data which are often annotated by humans. However, for most languages, large-scale labeled datasets are only readily available in some domains, for example the news domain. For other domains like social media and dialog texts, there is a lack of such data sets. The NER systems trained on the news domain often perform worse in other domains. It is a reasonable solution to create human-annotated data in new domains to improve the performance of NER system.

In this paper, we present a new corpus for Chinese Named Entity Recognition in multi-domains, named M-CNER. We create several data sets in three domains: human-computer interaction, social media, and e-commerce, which are often used in real applications. We require the annotators to label some predefined entities. In the annotation procedure, the annotators label the sentences independently in the first round. One sentence is labeled by more than one persons. In the second round, experts check the entities which have disagreement among the annotators. The detailed settings are described in Section 2.

Most traditional high performance sequence labeling models for NER are linear statistical models, including Hidden Markov Models (HMM) and Conditional Random Fields (CRF) (Ratinov and Roth, 2009; Passos et al., 2014; Luo et al., 2015). CRF has been widely used for this task for the last decades, but in the most recent years, non-linear neural networks have become popular for NER. For example, Collobert et al. (2011) propose a simple but effective feed-forward neutral network that independently assigns the NE labels for each word by using contexts within a window with fixed size. Recurrent neural networks (RNN) (Goller and Kuchler, 1996), together with its variants such as long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) and gated recurrent unit (GRU) (Cho et al., 2014), have shown great success in the task of NER. Among the above models, we choose three models including CRF, LSTM, and LSTM-CRF for comparisons.

In the experiments, we evaluate the three systems on the newly created data sets which are from different domains. The experimental results show that LSTM-CRF performs the best among the systems. The new corpus can be used for further studies in research community.

2. M-CNER Data

In this section, we describe how we create a corpus for Chinese Named Entity Recognition in Multi-domains, named M-CNER. We collect the sentences from three domains in Chinese: human-computer interaction (HCI), social media (SM), and e-commerce (ECO).

2.1 Annotation Procedure

In our annotation procedure, there are two rounds. In the first round, we hire several undergraduate students to perform annotation. They read guideline documents which describe the definitions of the predefined entity types. For each type, we additionally provide about 20 exemplifying sentences to help the annotators understand the definitions. Then they identify the named entities in the sentences and classify them as one of the predefined types. After the first round, there are some sentences for which the annotators give out different annotations. For those sentences, we let experts check the disagreed annotations carefully. Finally, the experts reach the agreements for all the cases with discussion.

2.2 Domains

2.2.1 Human-Computer Interaction (HCI)

In Human-Computer Interaction (HCI) domain, we collect raw sentences from an intelligent robotic company. And then we randomly select some sentences as our annotation

pool. We ask the annotators to label two types of entities: Person-name (PER) and Music-song (MUS). We create two data sets, HCI-PER for PER and HCI-MUS for MUS respectively. The annotators label the sentences independently and each sentence is assigned to three annotators. After annotation, we remove some illegal sentences reported by the annotators. Finally, we have 12,204 sentences for HCI-PER and 10,510 sentences for HCI-MUS.

Data set	#Sent	#Entity-type	#Entity-annotated
HCI-PER	12204	1	14359
HCI-MUS	10510	1	3646
SM-Weibo	3890	3	9534
ECO-Title	2323	5	10158
ECO-Query	2297	5	2665

Table 1: The information of M-CNER

Table 1 shows the information of annotated data, where #Sent refers to the number of sentences in the data set, #Entity-type refers to the number of prefined entity types, and #Entity-annotated refers to the number of entyties annoated in the sentences.

In the HCI dataset, HCI-PER includes: 1) full name, for example {习近平@PER}主席访美(President {Xi Jinping@PER} visited USA); 2) Surname+Title, for example {习主席@PER}访美({President Xi@PER} visited USA); 3) Musical ensemble, for example {羽泉@PER}({Yu-Quan@PER); 4) Nickname, for example 真甜啊，我的{甜心@PER }(You are so sweet, my {sweet-heart@PER}). HCI-MUS includes full song names.

2.2.2 Social Media (SM)

As for Social Media (SM) domain, the raw sentences are from the messages on Sina-Weibo (weibo.com). We apply the similar strategy as HCI to annotate the sentences. Three types of entities are defined: Person-name (PER), Orgnization-name (ORG), and Location (LOC). Peng and Dredze (2015) created a data set on Sina-Weibo, but the size is small. We additionally add 2,000 Weibo messages with the same entity definition. As for annotation guideline, we follow the definition of Peng and Dredze (2015). Totally, we get 3,890 messages for this domain as shown in Table 1. We treat one message as one sentence in the experiments.

2.2.3 E-Commerce (ECO)

As for E-Commerce (ECO) domains, we collect the sentences from an e-commerce plaform. The sentences are from two parts : one is titles of products (ECO-Title) and another is user queries (ECO-Query). We also use the similar annotation strategy for this domain. We seperate the sentences into two data sets : ECO-Title and ECO-Query, because the styles of sentences from two parts are quite different. Five types of entities are defined : brand, product, model, specifications, and material. Finally, we have 2,323 sentences for ECO-Title and 2,297 for ECO-Query as shown in Table 1.

We list some examples of ECO-Title and ECO-Query in Table 2, where we give one Title example and one Query example for each type.

2.3 Data Splits

For our experiments, we split the data into three parts : training, development, and test sets. Table 3 shows the detailed information of data splits for M-CNER. For HCI and ECO domains, we use the percentage 8 :1 :1 for three parts. As the SM data, we use newly annotated 2000 messages as training data, the data created by Peng and Dredze (2015) as development and test data.

Type	Examples
Brand	Title : {品胜@Brand}移动电源适用于{苹果@Brand}{华为@Brand }{OPPO @Brand} {Pingshen@Brand} mobile power for {Apple@Brand}, {Huawei@Brand}, and {OPPO@Brand} Query: {华为@Brand}和{荣耀@Brand}，哪个系列漂亮? {Huawei@Brand} and {Honer@Brand}, which series are beautiful?
Product	Title : 苹果 iphone8 全网通 4G{手机@product} Apple iphone8 Full Netcom 4G {mobile phone@Product} Query: 我想买牛肉味的{兰花豆@product} I want to buy beef flavor {orchid beans@Product}
Model	Title: 苹果{iphone8@Model}全网通 4G手机 Apple {iphone8@Model} Full Netcom 4G mobile phone Query: {iphone8@Model}比{iphone7 @Model}有哪些提升 Which features are {iphone8@Model} better than {iphone7@Model}
Material	Title: 英伦日常百搭【牛皮@Material】鞋子 British daily {cowskin@Material} shoes Query: 有亚麻裤子吗？ Do you have {flax@Material} pants?
Specif	Title: 镜片护理液{3 瓶@ Specif }{120ml @ Specif }装 lens care solution {3 bottles@Specif} {120ml@ Specif} Query: 我要买{三箱@ Specif }牛奶 I want to buy {three boxes@ Specif} of milk

Table 2 : Examples of ECO data

3. Comparison Approaches

In this paper, we compare the performance of three systems which are frequently used in the task of NER on the newly created data sets. The first one is a traditional system based on the CRF model (Lafferty et al., 2001), the other two are based on neural networks: Bi-LSTM without/with the CRF layer. We describe the models briefly since full details are presented in the related papers.

Domain	Train	Dev	Test
HCI-PER	10023	1114	997
HCI-MUS	8510	1000	1000
SM-Weibo	2000	890	1000
ECO-Title	1863	230	230
ECO-Query	1837	230	230

Table 3: The data splits of M-CNER

3.1 CRF

For sequence labeling (or general structured prediction) tasks, the performance can be improved by considering the correlations between labels in neighborhoods and the system jointly generates the best chain of labels for a given input sentence. For example, in the sequences with standard BIO2 schema (Tjong Kim Sang and Veenstra, 1999), I-ORG cannot follow I-PER. We build a NER system by using a conditional random field (CRF) model (Lafferty et al., 2001) which performs very well in the task of NER.

Formally, we use $\mathbf{x}=\{x_1, \cdots, x_n\}$ to represent a generic input sequence where x_i refers to the ith word. $\boldsymbol{y} = \{y_1, \cdots, y_n\}$ represents a generic sequence of labels for **x**. $\mathcal{Y}(\boldsymbol{x})$ denotes the set of possible label sequences for **x**. The probabilistic model calculates the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ over all possible label sequences **y** given **x** with the following form:

$$\mathrm{p(y|x)} = \frac{1}{Z(x)} \exp\left(\sum_{i,k} w_k f_k(y_{i-1}, y_i, x) + \sum_{i,k} w_k{}' f_k{}'(y_i, x)\right)$$

where Z(x) is the normalization constant, f_k is a binary feature function, and w_k is the weight of f_k. Given the training data, the parameters of the model are trained to maximize the conditional log-likelihood. In the testing stage, given a sentence x in the test data, the tagging sequence y* is given by,

$$y^* = \arg max_{y'} p(y'|x)$$

CRF allows us to utilize a large number of observation features as well as different state sequence based features and other features we want to add. For a sequence CRF model (only interactions between two successive labels are considered), training and decoding can be solved efficiently by adopting the Viterbi algorithm. As for the feature templates, we use the supervised version of Zhao et.al (2008).

3.2 Bi-LSTM

In this section, we introduce the system based on bi-directional without the CRF layer.

3.2.1 LSTM Unit

Recurrent neural networks (RNNs) are a powerful family of connectionist models that capture time dynamics via cycles in the graph. Though, in theory, RNNs are capable to capturing long-distance dependencies, in practice, they fail due to the gradient vanishing/exploding problems (Bengio et al., 1994; Pascanu et al., 2012).

LSTMs (Hochreiter and Schmidhuber, 1997) are variants of RNNs designed to cope with these gradient vanishing problems. Basically, a LSTM unit is composed of three multiplicative gates which control the proportions of information to forget and to pass on to the next time step.

Formally, the formulas to update an LSTM unit at time t are:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right)$$
$$\tilde{c}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$
$$h_t = o_t \odot \tanh(c_t)$$

where σ is the element-wise sigmoid function and $\odot$ is the element-wise product. x_t is the input vector (e.g. char embedding) at time t, and h_t is the hidden state (also called output) vector storing all the useful information at (and before) time t. U_i, U_f, U_c, U_o denote the weight matrices of different gates for input x_t, and W_i, W_f, W_c, W_o are the weight matrices for hidden state h_t. b_i, b_f, b_c, b_o denote the bias vectors. It should be noted that we do not include peephole connections (Gers et al., 2003) in the our LSTM formulation.

3.2.2 Bi-LSTM

For many sequence labeling tasks it is beneficial to have access to both past (left) and future (right) contexts. However, the LSTM's hidden state h_t takes information only from past, knowing nothing about the future. An elegant solution whose effectiveness has been proven by previous work (Dyer et al., 2015) is bi-directional LSTM (Bi-LSTM). The basic idea is to present each sequence forwards and backwards to two separate hidden states to capture past and future information, respectively. Then the two hidden states are concatenated to form the final output. We treat NER as a classification problem in the final stage.

3.3 Bi-LSTM-CRF

Then, we can add a CRF layer to the Bi-LSTM model as shown in Figure 1. That is Bi-LSTM-CRFs (Huang, Xu, and Kai 2015) which are well-suited for sequence labeling. Bi-LSTM-CRF can be regarded as a combination of bidirectional LSTM and CRF.

By contrast to the local classification, CRFs (Lafferty, McCallum, and Pereira 2001) have the advantage of modeling at the sentence level instead of individual

positions. Finally, we feed the output of Bi-LSTM into the CRF layer directly for NER decoding.

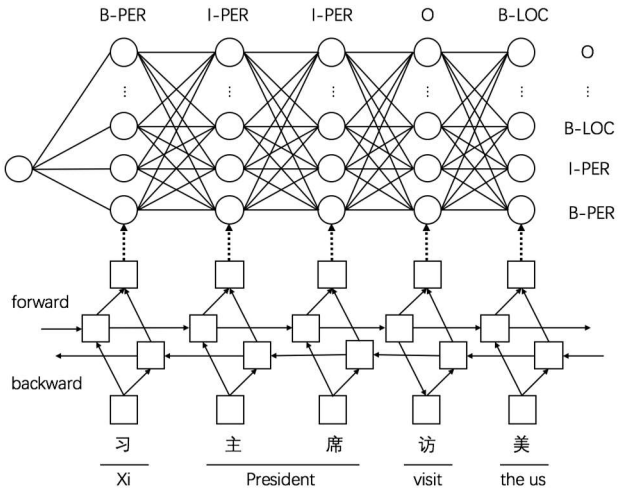

Figure 1: The framework of Bi-LSTM-CRF

3.4 Settings for Neural Networks

Lexical embeddings represent words in a continuous low dimensional space, which can capture semantic or syntactic properties of the lexicon. Similar words would have similar low dimensional vector representations. Embeddings have been used to gain improvements in a variety of NLP tasks. In NER specifically, several studies have shown improvements by using pre-trained neural embeddings as features in standard NER systems (Collobert and Weston, 2008; Turian et al., 2010; Passos et al., 2014). More recently, these improvements have been demonstrated on Twitter data (Cherry and Guo, 2015). Embeddings are especially helpful when there is little training data, since they can be trained on a large amount of unlabeled data.

However, training embeddings for Chinese is not straightforward: Chinese is not word segmented, so embeddings for each word cannot be trained on a raw corpus. Additionally, the state-of-the-art systems for downstream Chinese tasks, such as NER, may not use words. Thus, we use character embeddings for our systems instead.

3.4.1 Character Embeddings

We learn an embeddings for Chinese characters in the training corpus (Sun et al., 2014; Liu et al., 2014). This setting does not require pre-processing the text, and better fits our task studied in this paper: Chinese NER tagging over characters. Since there are many fewer characters than words in Chinese, we can reduce the size of embeddings. On the one hand, this means fewer parameters and less over-fitting. However, the reduction in parameters comes with a loss of specificity, where we may be unable to learn different behaviors of a character in different settings. We explore a compromise approach in the next section. The embeddings are directly incorporated into the NER system by adding embedding features for each character.

For each of the embeddings, we fine-tune pretrained embeddings in the context of the NER task. This corresponds to initializing the embeddings parameters using a pre-trained model, and then modifying the parameters during gradient updates of the NER model by back-propagation gradients. This is a standard method that has been previously explored in sequential and structured prediction problem (Collobert et al., 2011; Zheng et al., 2013; Yao et al., 2014; Pei et al., 2014).

3.4.2 Training Settings

To train model parameters, we exploit a negative log likelihood objective as the loss function. We apply softmax over all candidate output label sequences and use standard back-propagation method to minimize the loss function of the CRF model.

4. Experiments

In this section, we evaluate the three systems on M-CNER, our newly created data sets.

4.1 Evaluation Setup

The vector representations of characters are basic inputs of our systems based on neural networks, which are listed by the looking-up table. We use pretrained embeddings trained on large-scale raw corpus. For the systems on SM domain, we use a data downloaded from the site of Sina-Weibo, having 5M messages. And for the systems on HCI and ECO domains, we use a data from the user-generated content from Web, having 5M sentences. For training the embeddings, we use the word2vec in the experiments.

As for the hyper-parameters, we tune them on the development set. After tuning, we set the dimension size of character embeddings as 100, the dimension sizes of all the other hidden layers also as 100, the mini-batch size as 128, and the learning rate is 0.01. We adopt the dropout technique to avoid overfitting by a drop value of 0.2.

We report the scores by precision, recall, and F1 as the previous studies did.

4.2 Main Results

In this section, we show the model performances of the comparison systems. Table 4 shows the experimental results on all the datasets, where BM and BM-CRF refer to Bi-LSTM without/with the CRF layer, respectively.

From the table, we find that the CRF model provides better scores on precision than Bi-LSTM and Bi-LSTM-CRF while Bi-LSTM-CRF performs the best on F1 in the most cases except for SM-Weibo. In average, Bi-LSTM-CRF achieves better performance with absolute score +4.31% than CRF. This indicates that the recent neural networks models are more powerful for NER. The results of Bi-LSTM and Bi-LSTM-CRF also show that adding the CRF layer to Bi-LSTM is very important for improving the performance.

The information of Table 1 shows that the numbers of sentences in HCI-PER and HCI-MUS are much larger than the ones in SM-Weibo, ECO-Title and ECO-Query. The results from Table 4 show that the F1 scores are only on the level of 40-60% on SM-Weibo, ECO-Title and ECO-Query, while the scores are around 90% for HCI domains. We will label more sentences in SM and ECO domains to improve the performance further in future work.

Data	Model	Precision	Recall	F1
HCI-PER	CRF	**95.37**	79.14	86.5
	BM	83.18	83.57	83.37
	BM-CRF	90.8	**89.74**	**90.27**
HCI-MUS	CRF	**88.51**	77.74	83.89
	BM	73.90	80.80	77.20
	BM-CRF	87.77	**84.64**	**86.17**
SM-Weibo	CRF	**69.53**	40.19	**50.94**
	BM	32.38	31.99	32.18
	BM-CRF	50.54	**43.82**	46.94
ECO-Title	CRF	**71.81**	31.47	43.76
	BM	48.97	52.28	50.57
	BM-CRF	63.23	**54.99**	**58.82**
ECO-Query	CRF	**65.84**	47.32	55.06
	BM	45.99	**58.93**	51.66
	BM-CRF	61.03	58.04	**59.50**
Average	CRF	**78.21**	55.17	64.03
	BM	56.88	61.51	58.99
	BM-CRF	70.67	**66.25**	**68.34**

Table 4: Main results on M-CNER

5. Conclusion

In this paper, we have presented a new corpus for Chinese Named Entity Recognition in three domains, named M-CNER. The data sets are first labeled by the annotators and then reach the agreements via the discussion by the experts. We evaluate three popular systems on our newly created corpus. The experimental results show that Bi-LSTM-CRF performs better than the other two systems. This new corpus can be used as evaluation benchmark for research community and we can build better Chinese NER systems for the three domains.

In future work, we plan to add more sentences to SM and ECO domains, create more data set for other domains and define more types of entities. We will also build some state-of-the-art systems for comparisons on our data sets.

Acknowledgments

The research work is supported by the National Natural Science Foundation of China (61572338) and the Natural Science Foundation of the Jiangsu Higher Education Institutions(Grant No. 16KJA520001). The corresponding author is Wenliang Chen.

References

Lafferty J D, Mccallum A, Pereira F C N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data[C]. In ECML. Morgan Kaufmann Publishers Inc. 2001:282-289.

Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging[J]. Computer Science, 2015.

Bunescu R C, Mooney R J. A shortest path dependency kernel for relation extraction[C]. In EMNLP. Association for Computational Linguistics, 2005:724-731.

Jurafsky D, Martin J H. Speech and Language Processing (2nd Edition)[M]. Prentice-Hall, Inc. 2009.

Ratinov L, Roth D. CoNLL '09 Design Challenges and Misconceptions in Named Entity Recognition[C]. In CoNLL. 2009:147--155.

Passos A, Kumar V, Mccallum A. Lexicon Infused Phrase Embeddings for Named Entity Resolution[J]. Computer Science, 2014.

Luo G, Huang X, Lin C Y, et al. Joint Entity Recognition and Disambiguation[C]. In EMNLP. 2015:879-888.

Collobert R, Weston J, Karlen M, et al. Natural Language Processing (Almost) from Scratch[J]. Journal of Machine Learning Research, 2011, 12(1):2493-2537.

Goller C, Kuchler A. Learning task-dependent distributed representations by backpropagation through structure[C]. IEEE International Conference on Neural Networks. IEEE, 2002:347-352 vol.1.

Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735.

Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM[J]. Neural Computation, 2000, 12(10):2451.

Cho K, Merrienboer B V, Bahdanau D, et al. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches[J]. Computer Science, 2014.

Peng N, Dredze M. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings[C]. In EMNLP. 2015:548-554.

Sang E F T K, Veenstra J. Representing Text Chunks[J]. Computer Science, 1999:173--179.

Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[M]. IEEE Press, 1994.

Pascanu R, Mikolov T, Bengio Y. On the difficulty of training Recurrent Neural Networks[J]. Computer Science, 2012, 52(3):III-1310.

Gers F A, Schraudolph N N. Learning precise timing with lstm recurrent networks[M]. JMLR.org, 2003.

Dyer C, Ballesteros M, Ling W, et al. Transition-Based Dependency Parsing with Stack Long Short-Term Memory[J]. Computer Science, 2015, 37(2):321–332.

Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging[J]. Computer Science, 2015.

Zhao, H., and Kit, C. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition[C]. In IJCNLP, 106–111.