Proceedings of Machine Learning Research 95:518-533, 2018 ACML 2018

Character-based BiLSTM-CRF Incorporating POS and Dictionaries for Chinese Opinion Target Extraction

Yanzeng Li LIYANZENG@IIE.AC.CN
Tingwen Liu LIUTINGWEN@IIE.AC.CN
Diying Li LIDIYING@IIE.AC.CN
Quangang Li LIQUANGANG@IIE.AC.CN
Jinqiao Shi SHIJINQIAO@IIE.AC.CN
Yanqiu Wang WANGYANQIU@IIE.AC.CN
Institute of Information Engineering, Chinese Academy of Sciences
School of Cyber Security, University of Chinese Academy of Sciences

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Opinion target extraction (OTE) is a fundamental step for sentiment analysis and opinion summarization. We analyze the difference between Chinese and the Indo-European languages family, and reduce Chinese OTE to a character-based sequence tagging task. Then we introduce two novel features for each character by distributing POS differentially and using predefined templates over contexts and dictionaries. We further propose a character-based BiLSTM-CRF model incorporating the two feature sequences aligned with the character sequence. Experimental results on real-world consumer review datasets show that our work significantly outperforms the baseline methods for Chinese OTE.

Keywords: Opinion target extraction, opinion mining, sentiment analysis, information extraction, deep learning

1. Introduction

Opinion target extraction (OTE) is a fundamental task in opinion mining and sentiment analysis. The task aims at identifying the items which opinion is expressed on from text (Liu et al., 2012). Usually, the text is customer review and the items include products, services and so on. For example, in the sentence "Central Park is a good place for relaxation" express a positive opinion toward the target "Central Park". Due to its wide application range, OTE has been a popular research topic.

Prior works on OTE can be divieded into three categories, namely rule-based methods, linear statistical methods and deep learning methods. The rule-based methods extract opinion targets with predefined linguistic patterns including part-of-speech (POS), word dictionary, etc (Popescu and Etzioni, 2005; Qiu et al., 2011; Wang et al., 2014; Mukherjee and Liu, 2012). The linear statistical methods use hidden markov model (HMM), maximum entropy markov model (MEMM) (Mccallum et al., 1999) and conditional random field (CRF) (Lafferty et al., 2001) to extract opinion targets from reviews (Jin, 2009; Li et al., 2010; Ma and Wan, 2010; Jakob and Gurevych, 2010). These statistical models need a large number of hand-crafted features to work well. Recently, deep learning methods are proposed

to automatically extract high order features and achieve competitive performance in many sentiment analysis tasks. Liu et al. (2015b) apply recurrent neural networks (RNN) and word embeddings to extract opinion targets. Poria et al. (2016) introduce deep convolution neural networks (CNN) (Collobert et al., 2011) for the OTE task, and combine linguistic patterns to achieve better performance. A RNCRF (Wang et al., 2016) consists of recursive neural network (Socher et al., 2010) and CRF, which collectively extract opinion targets based on dependency trees.

Generally, prior works model the extraction problem as a word-based sequence tagging task, and most are designed for English OTE. However, Chinese is much different from the Indo-European languages family: in Chinese, words are not separated by explicit delimiters, and character is ideogram, morpheme and logograph. This means that character is the smallest morpheme unit in Chinese, and is also the linguistic unit that can express semantics. Thus, the problem of Chinese OTE should be modeled as a character-based sequence tagging tasks. Previous works (Popescu and Etzioni, 2005; Qiu et al., 2011; Zhang et al., 2018) show that incorporating features extracted from POS and word dictionary can improve the effectiveness of word-based sequence tagging tasks. Note that POS and dictionary are prepared for words, how to utilize them well in a character-based sequence tagging task is an open problem.

In this paper, we propose a character-based BiLSTM-CRF model for Chinese OTE. The BiLSTM (bidirectional long short-term memory) layer models the context information of each character. The hidden states of the BiLSTM layer are fed into the CRF layer to optimize sequence tagging with the help of adjacent tags. The BiLSTM-CRF model can work well for Chinese OTE.

To utilize POS, we propose a novel feature named [CP-POS]@C for each character, which distributes POS to all characters of the same word differentially. The main idea is use the character position (denoted as CP hereinafter) tag in the corresponding word to assist POS distribution among characters. To utilize dictionary, inspired by the work on Chinese word segmentation (Zhang et al., 2018), we introduce a novel feature named DictFeature for each character, whose value is dependent on word dictionary and character contexts using predefined templates. We further give a flexible framwork to incorporate [CP-POS]@C feature and DictFeature into our character-based BiLSTM-CRF model.

This paper makes the following three contributions. Firstly, to our knowledge, we are the first to address the Chinese OTE task with deep learning models. Specifically, we introduce a character-based BiLSTM-CRF model. Secondly, we design two novel character features extracted from POS and word dictionary, and incorporate them into our BiLSTM-CRF model in a flexible framework. Thirdly, we conduct plenty of experiments on three real-world datasets, and the results show that our proposed method can outperform baseline methods with a large margin. [1]

1. Our code and data are available on https://github.com/kdsec/chinese-opinion-target-extraction

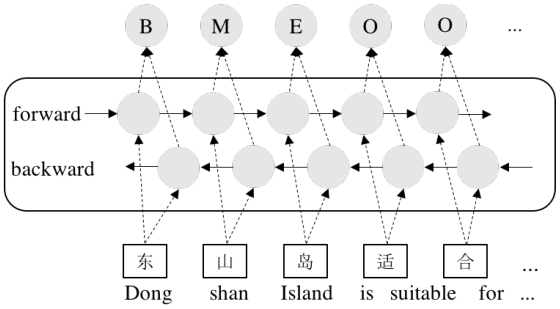

Figure 1: An illustration of BiLSTM network architecture

2. Preliminaries

2.1. Problem Statement

Given a sentence $\mathbf{x} = (x_1, x_2, ..., x_n)$, the result of sequence tagging $\mathbf{y} = (y_1, y_2, ..., y_n)$ is assigned to every character of the sentence. We decide to use IOBES (Inside, Outside, Beginning, End, Single) tagging scheme. Incorporating POS and dictionaries with a sequence tagging model can improve the performance for OTE task. The POS and dictionaries information is indirectly associate to each character. Hence, how to design features for each character from POS and word dictionary, and incorporate them into a sequence tagging model is an open problem to be addressed.

2.2. BiLSTM

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a variant of the RNN. This network architecture can efficiently solve the long-term dependencies problem by introducing gate mechanism and memory cell.

The LSTM cell can compute current hidden state $\mathbf{h}_t$ based on current vector $\mathbf{x}_t$, previous hidden state $\mathbf{h}_{t-1}$ and previous cell state $\mathbf{c}_{t-1}$. The operation of input gate i, forget gate f, output gate o and memory cell c is defined as:

$$\mathbf{i}_t = \delta\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i\right) \quad (1)$$

$$\mathbf{f}_t = \delta\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f\right) \quad (2)$$

$$\tilde{\mathbf{c}_t} = tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{c}_t} \quad (4)$$

$$\mathbf{o}_t = \delta\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_{t-1} + \mathbf{b}_o\right) \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t * tanh\left(\mathbf{c}_t\right) \quad (6)$$

where δ is the element-wise sigmoid function, $*$ is the element-wise product, $\mathbf{W}_{(\cdot)}$ are the weight metrics, and $b_{(\cdot)}$ are the biases.

For each vector $\mathbf{x}_t$, the hidden state $\mathbf{h}_t$ only gets past information. The bidirectional LSTM (BiLSTM) architecture (Gers et al., 2000) is used to capture both past and future information by concatenating hidden state $\overrightarrow{\mathbf{h}_t}$ of forward LSTM and $\overleftarrow{\mathbf{h}_t}$ of backward LSTM. So the hidden state of BiLSTM could be defined as:

$$\mathbf{h}_t = \overrightarrow{\mathbf{h}_t} \oplus \overleftarrow{\mathbf{h}_t} \quad (7)$$

Figure 1 gives an illustration of BiLSTM architecture for Chinese OTE.

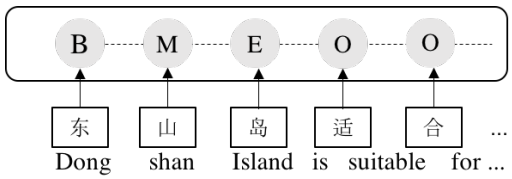

Figure 2: An illustration of CRF architecture

2.3. CRF

For sequence tagging task, it is important to consider the dependencies of adjacent labels. For example, I (Inside) tag cannot follow E (End) tag or S (Single) tag.

Therefore, Lafferty et al. (2001) adopt a conditional random field (CRF) to predict the tags of whole sentence jointly. CRF introduces a transition matrix $\mathbf{A}$, which measure the score from tag p to tag q by $\mathbf{A}_{p,q}$.

Formally, we use $\mathbf{x} = (x_1, x_2, ..., x_n)$ to represent an input sequence, the score of corresponding prediction tag sequence $\mathbf{y}$ can be defined as:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n+1} \left(\mathbf{A}_{y_{t-1}, y_t} + \mathbf{P}_{t, y_t} \right) \tag{8}$$

$\mathbf{P}_{t,y_t}$ measures the score of x_t tagging to y_t. The value of $\mathbf{P}_{t,y_t}$ depends on the usage situation. Considering all possible tag sequences, the probability of sequence $\mathbf{y}$ can be defined as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\prod^n e^{s(\mathbf{x},\mathbf{y})}}{\sum_{\widetilde{\mathbf{y}} \in \mathbf{Y_x}} \prod^n e^{s(\mathbf{x},\widetilde{\mathbf{y}})}} \tag{9}$$

where $\mathbf{Y_x}$ is all possible tag sequences of sentence $\mathbf{x}$. While training, we use maximum likelihood estimation to maximize the log-probability of correct tag sequence:

$$log\left(p(\mathbf{y}|\mathbf{x})\right) = s(\mathbf{x}, \mathbf{y}) - log\left(\sum_{\widetilde{\mathbf{y}} \in \mathbf{Y_x}} e^{s(\mathbf{x},\widetilde{\mathbf{y}})} \right) \tag{10}$$

While decoding, we use the tag sequence $\mathbf{y}^*$ with maximum score as prediction result:

$$y^* = \arg\max_{\widetilde{\mathbf{y}} \in \mathbf{Y_x}} s(\mathbf{x}, \widetilde{\mathbf{y}}) \tag{11}$$

After training, CRF could select the best tag sequence from all possible tag sequences. Figure 2 shows the CRF architecture for Chinese OTE.

3. Methodology

As mentioned above, for OTE task, we employ BiLSTM model to capture semantic information in sentence, and use CRF to ensure the correct order of tag sequence. In Chinese OTE, we tag character in the model rather than word to produce results, because the latter complicates the process of texts pretreatment and introduces errors to recognize entities

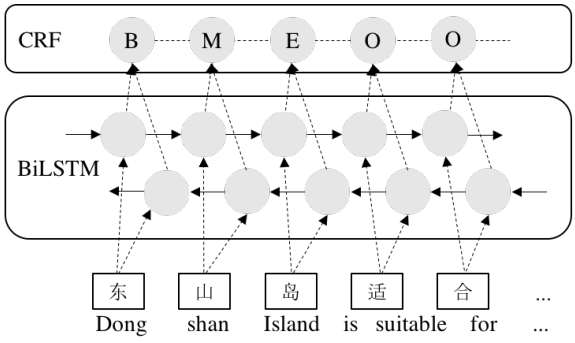

Figure 3: An illustration of BiLSTM-CRF architecture

boundaries. More narrowly, automatic word segmentation may cut entities into slices and the entities boundaries may connect more characters in the slices. However, based on our analysis of opinion targets, POS sequence information is useful in this task. Domain dictionaries are easy to acquire and can bring improvement to it. Both of them cannot directly incorporate to neural networks.

3.1. BiLSTM-CRF Model

As mentioned above, the BiLSTM layer is used to capture both past and future information, the CRF layer is used to predict the tags of whole sentence jointly by considering the dependencies of output tags. Therefore, we construct our neural network by using hidden state of BiLSTM layer as input sequence of CRF layer. The hidden state of BiLSTM could be defined as Equation (7) We can use the concatenation hidden states, denoted as $\mathbf{h}_t$ of x_t to predict the tag of corrected x_t directly, since it contains the context information. Then we feed those hidden states $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n)$ into CRF layer.

In particular, $\mathbf{P}$ in Equation (8) is:

$$\mathbf{P} = \mathbf{W}_s\mathbf{h} + \mathbf{b}_s \tag{12}$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are the parameters. The size of $\mathbf{P}$ is $n \times k$, where k is the number of tags. Meanwhile, the size of $\mathbf{A}$ in Equation (8) is $(k+2) \times (k+2)$ because we use additional *start* and *end* tags to represent y_0 and y_{n+1}.

In the training and decoding processes, we use bi-gram iterations of outputs, the dynamic programming algorithm therefore can be used to compute Equations (10) and (11).

In summary, the output vectors of BiLSTM are fed to CRF layer to jointly decode the best label sequence. Figure 3 gives the basic structure of BiLSTM-CRF model. Experimental results show that BiLSTM-CRF model significantly outperforms BiLSTM model.

3.2. [CP-POS]@C Feature Construction

Note that POS attribute is naturally associated with words rather than characters. Inspired by the way of combining tags for joint extraction of entities and relations (Zheng et al., 2017), we propose a feature named [CP-POS]@C that is extracted from POS attribute and character position information for each character.

Table 1: An example of obtaining [CP-POS]@C

Sentence	(Dongshan Island is suitable for self driving tour)							
WS/POS	东山岛/n			适合/v		自/r	驾游/v	
Char	东	山	岛	适	合	自	驾	游
POS@C	n	n	n	v	v	r	v	v
CP	B	M	E	B	E	S	B	E
[CP-POS]@C	B-n	M-n	E-n	B-v	E-v	S-r	B-v	E-v

Table 1 is give an illustration of how to obtain [CP-POS]@C with POS and CP. As shown in the table, for an input sentence, we acquire POS attributes and word segmentation at the same time by NLP tools. It is intuitive to directly distribute the POS attribute of a word into each character, which is called POS@C that ignores the position information of characters in a word. We want to distribute POS with CP information together. The word segmentation, which can be regarded as a sequence tagging problem, distributes a position tag for each character in words named CP. So we combine the character position tags with POS attribute of the corresponding word to build [CP-POS]@C feature for each character.

Each [CP-POS]@C feature includes two parts: the character position tag in corresponding word and the POS attribute. The CP tag is in the set $\{B, M, E, S\}$ which represent the "Beginning", "Middle", "End" and "Single" position of a word respectively. The POS attributes in different languages usually have different compositions of set. In this case, we mainly talk about the POS types of Chinese. Thus, the total number of [CP-POS]@C is $4 * N_{wp}$, where N_{wp} is the number of POS types. In this way, we are able to generate a [CP-POS]@C feature for each character, which distributes the POS attribute into each character of the word with character position information.

3.3. DictFeature Construction

As we know, consumers publish reviews of specific domains in Internet, which usually contain name lists of products or services. This can be used to build domain dictionary. We extract DictFeature for each character based on the built dictionary and predefined teature templates, inspired by the work on Chinese word segmentation (Zhang et al., 2018).

The method to construct DictFeature for a character consists of three steps. The first is to segment the context of a character based on feature templates. Table 2 shows the template information of x_t, when the given sentence is $\mathbf{x} = (x_1, x_2, ..., x_n)$. In fact, the templates are n-grams ($n \in [2, 5]$) containing x_t based on the context. For each character, we acquire its contexts and obtain 8 text segmentations according to templates. Then we generate a vector based on these segmentations and dictionaries. It is constructed to a vector of size 8 for each character. Each dimension of vector indicates whether the corresponding text segmentation is in dictionaries or not. Finally, we convert the vector to a number in particular range. The number range is from 0 to 255, when the vector dimension is 8. So we obtain DictFeature for each character, which is a particular number based on the context and dictionaries.

Table 2: Feature templates of DictFeature for the t-th character in a given sentence $\mathbf{x} = (x_1, x_2, ..., x_n)$

Type	Templates
2-gram	$x_{t-1}x_t, x_tx_{t+1}$
3-gram	$x_{t-2}\dots x_t, x_t\dots x_{t+2}$
4-gram	$x_{t-3}\dots x_t, x_t\dots x_{t+3}$
5-gram	$x_{t-4}\dots x_t, x_t\dots x_{t+4}$

3.4. Incorporating Features to BiLSTM-CRF

We propose a flexible framework to incorporate [CP-POS]@C and DictFeature to BiLSTM-CRF, as shown in Figure 4. The incorporating way is described step by step below.

3.4.1. Transferring Features to Vectors

As mentioned above, for a given sentence $\mathbf{x}$, we can construct [CP-POS]@C feature and DictFeature for each character. The character itself, [CP-POS]@C feature and DictFeature can be transferred into vectors by a embedding layer. The embedding layer can convert an element with one-hot representation to an embedding vector. As a result, we can get three vector sequences can, referred to as $\mathbf{x}^c$, $\mathbf{x}^p$ and $\mathbf{x}^d$, which indicate the sequence of character embedding, the sequence of [CP-POS]@C embedding, the sequence of DictFeature embedding respectively. Note that our proposed [CP-POS]@C feature is a distribution of POS with position information. Hence $\mathbf{x}^p$ contains rich information to understand the corresponding sentence. As mentioned above, BiLSTM layer can effectively capture context information of sequences. So we consider introducing a BiLSTM layer to utilize the [CP-POS]@C feature. The [CP-POS]@C embedding sequence is denoted as $\mathbf{x}^p = (\mathbf{x}_1^p, \mathbf{x}_2^p, ..., \mathbf{x}_n^p)$, where $\mathbf{x}_t^p$ is the [CP-POS]@C embedding of the t-th character. Feeding $\mathbf{x}_t^p$ into BiLSTM can produce 2 groups of hidden states: $\overrightarrow{\mathbf{h}_t^p}$ and $\overleftarrow{\mathbf{h}_t^p}$, which could express high-level characteristics of [CP-POS]@C.

3.4.2. Concatenation Vectors

After transferring features into vectors, for a given sentence $\mathbf{x}$, we obtain character embedding $\mathbf{x}_t^c$, two hidden state vectors of [CP-POS]@C embedding: $\overrightarrow{\mathbf{h}_t^p}$ and $\overleftarrow{\mathbf{h}_t^p}$, and DictFeature embedding $\mathbf{x}_t^d$ for character x_t. Different vectors can provide different aspects characteristics of a character. We also want to employ BiLSTM layer to capture the sequence information. As shown in Figure 4, we concatenate them to get the final input vector $\mathbf{x}_t$ of BiLSTM-CRF model, which is defined as:

$$\mathbf{x}_t = \mathbf{x}_t^c \oplus \overrightarrow{\mathbf{h}_t^p} \oplus \overleftarrow{\mathbf{h}_t^p} \oplus \mathbf{x}_t^d \tag{13}$$

It is noteworthy that this is not the only way to generate the the final input vector $\mathbf{x}_t$. For exampel, as mentioned above, we also obtain a [CP-POS]@C embedding vector $\mathbf{x}_t^p$, which contains more initial information than hidden state vectors: $\overrightarrow{\mathbf{h}_t^p}$ and $\overleftarrow{\mathbf{h}_t^p}$. Instead of

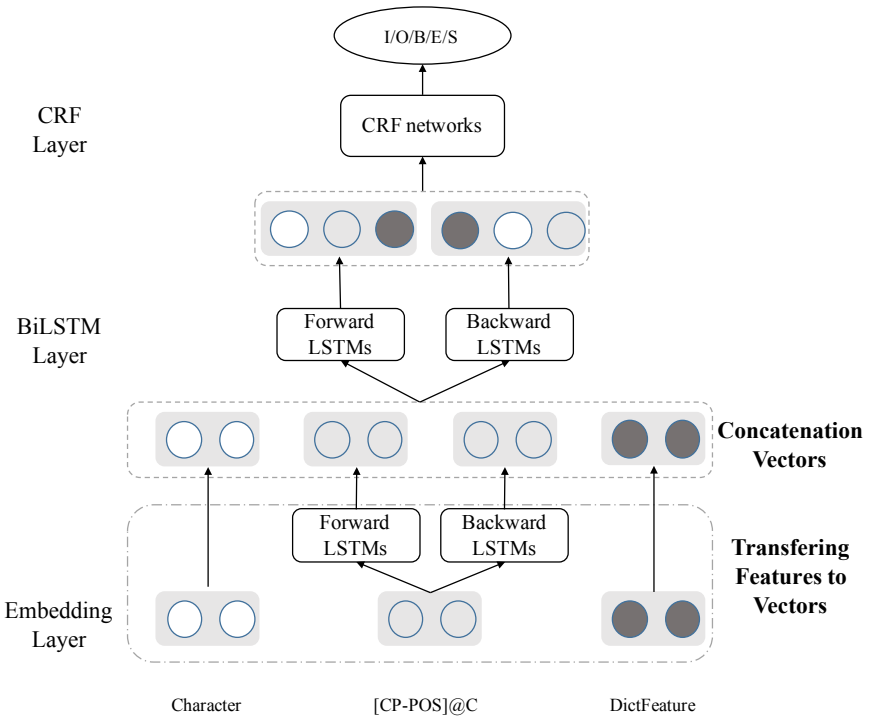

Figure 4: Our character-based BiLSTM-CRF model incorporating [CP-POS]@C feature and DictFeature.

concatenating $\overrightarrow{\mathbf{h}_t^p}$ and $\overleftarrow{\mathbf{h}_t^p}$, the [CP-POS]@C embedding vector can be used to concatenate with $\mathbf{x}_t^c$ and $\mathbf{x}_t^d$ to get the final input vector $\mathbf{x}_t$. We evaluate the effectiveness of different concatenation strategies in our experiments.

Then the final input vector $\mathbf{x}_t$ can be used as inputs of character-based BiLSTM-CRF. The BiLSTM layer can effectively acquire hidden states vectors, which comprehensively different aspect information of sequence $\mathbf{x}$. The operations are the same with Equations (2)-(6). The forward and backward hidden states are concatenated directly into a vector. Based on the hidden state vector sequence and transition information of tags, the CRF layer produce tag sequence. Both the training and encoding processes are described in Section 2.3 (see Equations (8)-(11)). In this way, we incorporate the [CP-POS]@C feature and DictFeature into character-based BiLSTM-CRF in a flexible way.

4. Experiments

4.1. Datasets

We conduct our experiments on three real-world datasets named Baidu, MaFengWo and Dianping respectively, which are collected or crawled from three well-known Chinese Internet companies, as the corresponding dataset name indicates. Each record consists of a

Table 3: Detailed information of our experimental datasets.

Datasets	Total	Train	Test	Types of Opinion Target
Baidu	12191	8533	3658	attractions, movies etc.
MaFengWo	58934	41253	17681	tourist attractions
DianPing	36083	25258	10825	restaurants

sentence and the corresponding opinion targets that appear in the sentence. We give a brief description for each dataset in the next.

Baidu: Baidu[2] held the second big data contest in 2016, designed for mining the factors influencing consumer decision. The contest task is just extracting opinion targets from consumer reviews. The contest opened 12 thousands of sentences with labeled opinion targets to the contestants, which constitute our Baidu dataset.

MaFengWo: MaFengWo[3] is the largest travel sharing website in China, providing global travel strategy, travel reviews and other integrated services. MaFengWo sets up an introduction page for each well-known attraction, where consumers can share their reviews. Thus, it is very likely that the attraction name is the opinion target of consumer reviews in the corresponding introduction page. We collect these reviews that are more than 15 characters and contain the corresponding attraction name. Finally we revise all the collected data by manual to build the MaFengWo dataset.

DianPing: DianPing[4] is China's leading O2O (Online to Offline) platform, which provides independent consumer reviews on local services. We get the DianPing dataset in the same way as constructing the MaFengWo dataset.

We divided each dataset into two parts: 70% records are used for training and 30% for testing. Detailed information of our experimental datasets are shown in Table 3. Note that opinion targets in the Baidu dataset do not belong to the same type, which may be tourist attractions, hospitals, movies, automobiles and so on.

4.2. Experimental Setting

Evaluation Metrics: We use precision, recall and F1-measure as the evaluation metrics to evaluate the performance of opinion target extraction. These three metrics are also widely-used in prior OTE works. We use the regular expression "[BIES]*" to match the output sequence in the greedy mode, and each matched segment is regarded as an opinion target. Note that in fact the regular expression relaxes the constraint condition of choosing segments, because strong constraints will make it impossible for us to find any segment as an opinion target.

Experimental Configurations: As shown in Figure 4, our neural network architecture mainly consists of embedding, BiLSTM and CRF layers. In the embedding layer, each character, its [CP-POS]@C and DictFeature are embedded into vectors of size 50, 20 and 20 respectively. Then it outputs 40 hidden states vector by using BiLSTM to operate the

2. https://baidu.com/
3. https://www.mafengwo.cn/
4. https://www.dianping.com/

Table 4: Comparision with baseline methods over three datasets.

Methods	Baidu			MaFengWo			DianPing		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
W-BiRNN	42.978	51.203	46.732	16.318	24.976	19.739	2.309	1.792	2.018
W-BiLSTM	43.627	50.437	46.786	16.678	26.882	20.585	2.799	2.882	2.840
C-BiRNN	67.273	51.586	58.394	60.937	51.773	55.983	42.262	29.718	34.897
C-BiLSTM	69.688	53.800	60.722	56.152	48.889	52.269	24.177	28.379	31.414
C-BiLSTM$^+$	63.027	53.171	57.681	40.555	77.552	53.259	63.909	57.580	60.579
Ours	**84.533**	**79.934**	**82.169**	**84.841**	**87.552**	**86.175**	**85.516**	**83.178**	**84.331**

[CP-POS]@C embedding. After concatenating, we get a vector of size 110. The BiLSTM layer yields 50 hidden states by using concatenation vector as inputs. In our experiments, the epoch, batch size and the learning rate are set to 15, 128 and 0.015. All the experiments in this paper are performed on two Nvidia Tesla P100 GPUs.

We build a Pytorch based framework, which uses BiLSTM-CRF with character embedding, DictFeature embedding, [CP-POS]@C hidden states vectors as inputs. All of sentences are padded and truncated to a fixed char-length of 250. THULAC (Sun et al., 2016) is employed to get [CP-POS]@C. We build our dictionary from BaiduBaike[5] and MaFengWo, which is independent to datasets.

Baseline Methods: We compare our work with word-based Bi-Elman-RNN (BiRNN) and word-based BiLSTM-RNN (BiLSTM) that were proposed in Liu et al. (2015a) and got well-performance for the opinion target extraction task in *SemEval-2014*. Incorporating POS attribute into the two models will give better performance as shown in Liu et al. (2015a). We further design three character-based models as the baseline methods to make a fair comparison. Detailed information of our five baselines are follows.

W-BiRNN: word-based BiRNN, with the concatenation of word embedding and POS embedding as inputs, as shown in Liu et al. (2015a).

W-BiLSTM: word-based BiLSTM, with the same inputs as W-BiRNN, as shown in Liu et al. (2015a).

C-BiRNN: character-based BiRNN, with the concatenation of character embedding and POS@C embedding as inputs.

C-BiLSTM: character-based BiLSTM, with the same inputs as C-BiRNN.

C-BiLSTM$^+$: character-based BiLSTM, with the concatenation of character embedding, [CP-POS]@C embedding and DictFeature embedding as inputs.

4.3. Experimental Results

4.3.1. Comparing with Baseline Methods

Table 4 presents experimental results of five baseline methods and our work that is a character-based BiLSTM-CRF model with same inputs as C-BiLSTM$^+$ here. We can draw three conclusions from the table.

5. https://baike.baidu.com/

Table 5: Experimental results of character-based BiLSTM-CRF models.

Inputs	Baidu			MaFengWo			DianPing		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
char baseline	85.753	77.228	80.358	78.730	83.485	81.038	78.934	82.901	80.869
+CP	81.971	79.798	80.870	79.377	83.072	81.183	80.866	82.106	81.481
+POS@C	82.978	78.759	80.813	81.311	81.127	81.219	81.210	81.811	81.509
+[CP-POS]@C	84.744	78.814	81.671	80.215	82.575	81.378	80.167	83.270	81.689
+BiLSTM-[CP-POS]@C	82.111	80.809	81.455	80.333	84.209	82.226	83.416	82.337	82.873
+DictFeature	**86.254**	**81.821**	**83.979**	**84.682**	**87.014**	**85.832**	**84.808**	**84.314**	**84.560**

Firstly, for Chinese OTE task, character-based models outperform word-based models. As shown in the labelled results, word-based models give too many terms as opinion targets, and some terms are even not meaningful Chinese words. This implies that words do not express semantics well for Chinese OTE, and errors in word-segmentation task would accumulate, causing poor model performance. The performance gap is much big especially on the DianPing and MaFengWo datasets, which have long opinion targets.

Secondly, incorporating the newly proposed [CP-POS]@C feature and DictFeature can improve the performance of character-based models sharply, comparing the results of C-BiLSTM$^+$ with that of C-BiLSTM.

Thirdly, as shown in the last two lines of Table 4, character-based BiLSTM-CRF gives higher performance than character-based BiLSTM, which demonstrates that our previous analysis is correct.

4.3.2. Effectiveness Analysis

We analyze the effectiveness of concatenation strategies and our proposed features for Chinese OTE, and show the results in Table 5. The first row named "char" gives a basic result, which use character embedding as the inputs of character-based BiLSTM-CRF. The next three rows show the results of concatenating character embedding with "CP" embedding, "POS@C" embedding and "[CP-POS]@C" embedding respectively. And the "char+BiLSTM-[CP-POS]@C" row gives the result of concatenating character embedding and the hidden states output by BiLSTM with the [CP-POS]@C as inputs. The result of our proposed method is shown in the last row.

[CP-POS]@C Feature: comparing with the "char" row, "char+[CP-POS]@C" achieves apparent improvement, which is in line with our previous analysis of the characteristics of opinion targets. In particular, we can see that both CP and POS@C can add some auxiliary information to the neural networks and get better results. With the help of the [CP-POS]@C feature we constructed, the comprehensive information of words including segmentation and POS are introduced to BiLSTM-CRF model to generate the prediction tags.

BiLSTM over [CP-POS]@C embedding: instead of using the concatenation of character embedding and [CP-POS]@C embedding as inputs directly, we use BiLSTM to get hidden states from [CP-POS]@C embedding to capture the context information. In this way, both MaFengWo and DianPing datasets show better results than the direct way. However, the F1 score of the Baidu dataset reduces slightly. We guess that the sentences and opinion targets of multiple domains in the Baidu dataset may be responsible for the

Table 6: Coverage ratios of dictionary on different datasets.

Datasets	Coverage in Train	Coverage in Test
Baidu	89.792	89.721
MaFengWo	98.444	98.535
DianPing	48.377	47.695

slight reduction. To make a long story short, extracting the context information by the BiLSTM is an effective method to exploit [CP-POS]@C feature.

DictFeature: obviously we get a significant improvement by adding dictionary information. As described above, the domain dictionary can be got from the Internet. For a given experimental dataset, the coverage ratio of dictionary represents what proportion of opinion targets in the dataset exists in the dictinary. Theoretically, we may get better results if the dictionary contains more opinion targets. The coverage ratios of dictionary for our three experimental datasets are shown in Table 6. We can see that DianPing dataset has the minimum coverage ratio and MaFengWo dataset has the maximum one among our three experimental datasets. Correspondingly, we get the lowest F1 score improvement (1.687%) on DianPing dataset and the highest improvement (3.606%) on MaFengWo dataset, comparing the results of row 6 with that of row 5 in Table 5.

5. Related Work

As a fundamental task in opinion mining and sentiment analysis, there are many works about OTE. Some works tackling the task focus on linguistic patterns such as POS and dictionaries. Popescu and Etzioni (2005) propose a method to determines whether a noun/noun phrase is an opinion target in product reviews through a Web search. Qiu et al. (2011) use double propagation (DP) to utilize syntactic relations between opinion targets and words. And the opinion dictionary is necessary to start the process, similar to (Wang et al., 2014; Mukherjee and Liu, 2012). These linguistic patterns are predefined and not flexible for different types of opinion targets. Sequence tagging is a classical task in natural language processing, which could solve tasks like speech recognition, POS tagging chunking, named entity recongition (NER), and semantic role labeling (SRL). Some works model the OTE problem as a sequence tagging task, and use traditional linear statistical models such as HMMs or CRFs to solve it (Jin, 2009; Li et al., 2010; Ma and Wan, 2010; Jakob and Gurevych, 2010). However, these liner statistical methods need many hand-crafted features. Some works use word alignment models to extract opinion targets from reviews (Liu et al., 2012, 2013a,b) and require a large number of data.

Recently, non-linear deep learning models are proposed to solve opinion mining and sentiment analysis problem because they can automatically learn features from data. Liu et al. (2015b) apply recurrent neural networks and word embedding to extract opinion targets. Yin et al. (2016) improve word embedding based on an unsupervised learning of distributed representations of words and dependency paths, and then use the improved word embedding as inputs to CRF. Li and Lu (2017) employed CRF and introduced a notion

of sentiment scope, performing well on named entity prediction and sentiment analysis. Poria et al. (2016) introduce deep convolution neural networks (Collobert et al., 2011) and combine linguistic patterns to achieve better performance. A RNCRF (Wang et al., 2016) consists of recursive neural network (Socher et al., 2010) and CRF, which collectively extract opinion targets based on dependency tree. A LSTM-based deep multi-task learning framework (Li and Lam, 2017) also jointly extract opinion targets from user reviews. An end-to-end multi-layer attention model (Wang et al., 2017) without requiring any parsers or linguistic resources can achieve competitive results.

BiLSTM (Gers et al., 2000), which captures forward and backward context information simutaneously, can achieve competitive performance against traditional models in sequence tagging tasks (Graves et al., 2013; Ling et al., 2015). Lample et al. (2016) proposed a BiLSTM-CRF model for NER, which can consider the dependencies of adjacent tags. We are the first to introduce the BiLSTM-CRF model to the OTE task.

For Chinese OTE, we also need to consider the characteristics of Chinese. One of the most important characteristics is that the boundaries of words are not clearly. Word segmentation is a necessary step of high-level analysis of Chinese texts. Peng and Dredze (2015) improved NER for Chinese social medias with word segmentation. Zhang et al. (2018) proposed a method to incorporate dictionaries for Chinese out-of-vocabulary words. Different word and character embedding methods are proposed for Chinese (Chen et al., 2015; Qiu et al., 2014). In Chinese sequence tagging tasks, prior methods usually label characters to avoid word segmentation errors (Dong et al., 2016).

6. Conclusion

In this paper, we propose an effective solution to extract opinion targets from Chinese consumer reviews, which is reduced to a character-based sequence tagging problem. Based on the analysis of the Chinese OTE task, we try to introduce POS attribute and word dictionary into a character-based BiLSTM-CRF model. To incorporate POS, we propose [CP-POS]@C feature which distributes POS with position information for each character. To incorporate dictionaries, we propose DictFeature based on character context information and word dictionary. Then a flxible framework is employed to incorporate the proposed two features into our character-based BiLSTM-CRF model. Experimental results show the effectiveness of our proposed method, which get 85% precision and 84% recall on average in real-world datasets.

References

A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 465–472, 2009.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huan-Bo Luan. Joint learning of character and word embeddings. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1236–1242, 2015.

Ronan Collobert, Jason Weston, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12(1):2493–2537, 2011.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition. In *International Conference on Computer Processing of Oriental Languages*, pages 239–250, 2016.

Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech Recognition with Deep Recurrent Neural Networks. *CoRR*, abs/1303.5778, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1035–1045, 2010.

John D. Lafferty, Andrew Mccallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270, 2016.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Yingju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 653–661, 2010.

Hao Li and Wei Lu. Learning latent sentiment scopes for entity-level sentiment analysis. In *AAAI*, pages 3482–3489, 2017.

Xin Li and Wai Lam. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892, 2017.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on*

Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 1520–1530, 2015.

Kang Liu, Liheng Xu, and Jun Zhao. Opinion target extraction using word-based translation model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1346–1356, 2012.

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. Opinion target extraction using partially-supervised word alignment model. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2134–2140, 2013a.

Kang Liu, Liheng Xu, and Jun Zhao. Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1754–1763, 2013b.

Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, 2015a.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1433–1443, 2015b.

Tengfei Ma and Xiaojun Wan. Opinion target extraction in chinese news comments. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 782–790, 2010.

Andrew Mccallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc of Icml*, pages 591–598, 1999.

Arjun Mukherjee and Bing Liu. Aspect extraction through semi-supervised modeling. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 339–348, 2012.

Nanyun Peng and Mark Dredze. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 548–554, 2015.

Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 339–346, 2005.

Soujanya Poria, Erik Cambria, and Alexander F. Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl.-Based Syst.*, 108:42–49, 2016.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27, 2011.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. Co-learning of word representations and morpheme representations. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 141–150, 2014.

Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, volume 2010, pages 1–9, 2010.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Li. Thulac: An efficient lexical analyzer for chinese. 2016.

Tao Wang, Yi Cai, Ho fung Leung, Raymond Y.K. Lau, Qing Li, and Huaqing Min. Product aspect extraction supervised with online domain knowledge. *Knowledge-Based Systems*, 71:86 – 100, 2014.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 616–626, 2016.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3316–3322, 2017.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2979–2985, 2016.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. Neural networks incorporating dictionaries for chinese word segmentation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, February 2-7, 2018, the Hilton New Orleans Riverside, New Orleans, Louisiana, USA.*, 2018.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1227–1236, 2017.