Supplementary Material for “A Neural Multi-digraph Model for Incorporating Gazetteers in Chinese NER”

Ruixue Ding¹, Pengjun Xie¹, Xiaoyan Zhang², Wei Lu³, Linlin Li¹ and Luo Si¹
¹Alibaba Group
²Beihang University, China
³Singapore University of Technology and Design
{ada.drX,chengchen.xpj,linyan.lll,luo.si}@alibaba-inc.com
xiaoyan.loic@gmail.com, luwei@sutd.edu.sg

Abstract
We present the e-commerce dataset information as well as gazetteers used in our model. The details of experiments are further discussed.

1 E-commerce Dataset
The E-commerce dataset is created by crawling and annotating product titles from the Taobao which is a Chinese e-commerce site with various types of products. Entity types including the Product name and the Brand name. Details of this dataset are shown in 1 and in 2.

2 Gazetteers
For general gazetteers, we collect gazetteers of 4 categories (PER, GPE, ORG, LOC). Each category has 3 gazetteers with different sizes, selected from multiple sources including Sougou (https://pinyin.sogou.com/dict/), HanLP (https://github.com/hankcs/HanLP) and Hankcs (http://www.hankcs.com/nlp/corpus). Sougou is a popular Chinese IME with a crowd source platform containing a huge number of gazetteers. HanLP is a widely used open-source Chinese NLP toolkit with many lexicons provided. Hankcs provides collection of lexicons of a ten million level volume.

For domain-specific gazetteers, We collect a list of person names from Weibo which is a Chinese microblog site. The gazetteers in the e-commerce domain are obtained by crawled product catalogues from Taobao.

3 Experimental Details
3.1 Hyper-parameter tuning
As shown in Table 3, parameters of NCRFPP are tuned on the OntoNotes development set by grid-search without gazetteers. We setup our model and compared models with the same configuration. The parameters of graph embedding are tuned on the OntoNotes development set by grid-search with one ORG gazetteer added.

3.2 Models for comparison
Wang et al. (2018) propose detailed description for constructing the following methods. We follow the same constructing method as them. These methods are the same as (Qi et al., 2019; Chiu and Nichols, 2016).

N-gram Given the input sentence $S$ with characters $c_1, \ldots, c_n$, the feature $f_{c_i}$ of $c_i$ is composed of 0-1 vectors (i.e., each entry of such vectors is either 0 or 1) for forward N-grams segments (e.g., $c_i c_{i+1}, c_i c_{i+1} c_{i+2}, \ldots$) and 0-1 vectors for backward N-grams segments (e.g., $c_{i-1} c_i c_{i-2}, \ldots$). The 0-1 vector indicates whether the segment can be found in gazetteers of a certain category (PER, GPE, ORG, LOC). For example, if $c_i c_{i+1}$ can be found in a PER gazetteer and a ORG gazetteer, its 0-1 vector should be

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char emb size</td>
<td>200</td>
<td>Learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Bigram emb Size</td>
<td>200</td>
<td>Batch size</td>
<td>10</td>
</tr>
<tr>
<td>LSTM hidden</td>
<td>600</td>
<td>Graph state</td>
<td>300</td>
</tr>
<tr>
<td>LSTM layers</td>
<td>2</td>
<td>T steps</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: The Entity Information

<table>
<thead>
<tr>
<th>Utterances</th>
<th>Tokens</th>
<th>Avg. Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>3989</td>
<td>2956</td>
</tr>
<tr>
<td>Test</td>
<td>498</td>
<td>1706</td>
</tr>
<tr>
<td>Dev</td>
<td>500</td>
<td>1685</td>
</tr>
</tbody>
</table>

Table 2: Statistics of Dataset

Table 3: Hyper-parameter values
Finally, \( f_{c_i} \) is the concatenation of all these 0-1 vectors.

**PIET** Given a sentence \( X \) and a gazetteer \( G \), we first select non-overlapping matches entities in segment \( X \) by maximizing the total number of matched tokens in \( X \). Then each character \( x_i \) is labeled as the gazetteer of the entity which \( x_i \) belongs to. The feature can be further represented in the format of one-hot encoding or feature embedding.

**PDET** PIET feature only considers the type of the entity which a character belongs to. Different from PIET feature, PDET feature also takes the position of a character in an entity into account: If the character is merely a single-character entity, we add a flag `S` before the PIET feature. Otherwise, for the first character of an entity, we add a flag `B` before the PIET feature; For the last character of an entity, we add a flag `E` before the PIET feature; For the middle character(s) of an entity, we add a flag `I` before the PIET feature. Similar to PIET feature, PDET feature can also be represented in the format of one-hot encoding or feature embedding.

References

