命名实体识别

Named Entity Recognition

胡伟龙
huweilong@whu.edu.cn

2018年6月15日

武漢大學 WUHAN UNIVERSITY | 计算机学院 Computer School of Wuhan University

Contents

INTRODUCTION

实体识别

- 早期工作主要针对专有名词：
  1. 人名(PERSON)——政治家、演员等
  2. 地名(LOCATION)——城市、州、国家等
  3. 机构名称(ORGANIZATION)
- 近年实体类型针对领域发生变化：
  1. 蛋白质、DNA、RNA和细胞类型等
  2. 药品和化学名称
  3. 产品和事件、物质、动物……

例子

国务院(机构名)总理李克强(人名)调研上海外高桥(地名)时提出，支持上海(地名)积极探索新机制。

命名实体识别方法

早期的研究大多采用基于人工构造规则的方法,而现在大多使用监督的机器学习方法。

1. 监督学习
    - 隐马尔科夫模型、决策树、最大熵模型、支持向量机、条件随机场
2. 半监督学习
    - "bootstrapping"方法，提供少量标注数据，搜索上下文信息
3. 无监督学习
    - 根据上下文的相似性从聚类组中收集命名实体

命名实体识别与序列标注

实体识别可以简单理解为序列标注问题：给定一个句子，为每一个字做标注。

序列标注实例

标注时使用IOB标注集:

习	近	平	视	察	了	湖	北	武	汉	。
B-PER	I-PER	I-PER	O	O	O	B-LOC	I-LOC	I-LOC	I-LOC	O

标注时使用IOBES标注集:

习	近	平	来	汉	视	察	。
B-PER	I-PER	E-PER	O	S-LOC	O	O	O

BiLSTM+CRF

BiLSTM+CRF

Interesting

There has been a running joke in the NLP community that an LSTM with attention will yield state-of-the-art performance on any task.

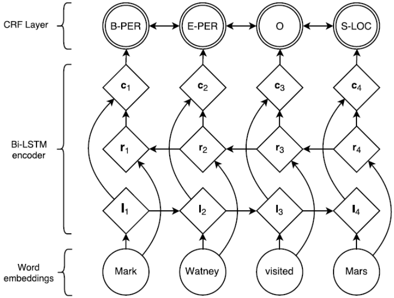
——Ruder.《Deep Learning for NLP Best Practices》

Neural Architectures for Named Entity Recognition

——arXiv2016

- **Motivation**
  - Language-specific resources and features are costly to develop in new languages and new domains.
  - Previous work have used unsupervised features to augment, rather than replace, hand-engineered features.
- **Two models**
  1. A bidirectional LSTM with a sequential conditional random layer above it.
  2. A new model that constructs and labels chunks of input sentences using an algorithm inspired by transition-based parsing.

Neural Architectures ... Recognition LSTM



$$i_t = \sigma(w_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$c_t = (1 - i_t) \odot c_{t-1}$$

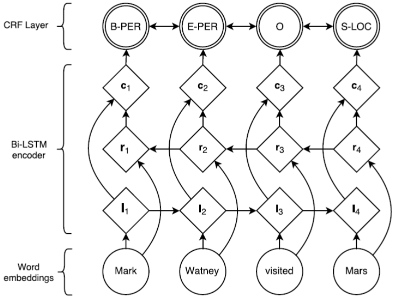
$$+i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \odot tanh(c_t)$$

LSTM

For a given sentence $(x_1, x_2, \ldots, x_n)$, the representation of word t is obtained by concatenating its left and right context representations, $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$



Sentences and predictions:

$$X = (x_1, x_2, \ldots, x_n)$$

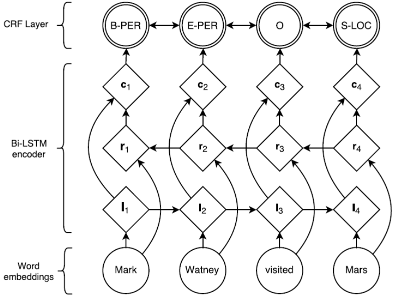
$$y = (y_1, y_2, \ldots, y_n)$$

The score is defined as follows:

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

CRF Tagging Models

- $P_{i,j} \in \mathbb{R}^{n \times k}$: the score of the j^{th} tag of the i^{th} word.
- $A_{i,j} \in \mathbb{R}^{k+2 \times k+2}$: the score of a transition from the tag i to tag j.



Probability for the prediction y:

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\widetilde{y} \in Y_X} e^{s(X,\widetilde{y})}}$$

Traning and decoding objective:

$$\log(p(y|X)) = s(X,y) - \log(\sum_{\widetilde{y} \in Y_X} e^{s(X,\widetilde{y})})$$

$$y^* = \arg\max_{\widetilde{y} \in Y_X} s(X, \widetilde{y})$$

Y_X represents all possible tag sequences for a sentence X.

Neural Architectures … Recognition Experiments

Model	F_1
Collobert et al. (2011)*	89.59
Lin and Wu (2009)	83.78
Lin and Wu (2009)*	90.90
Huang et al. (2015)*	90.10
Passos et al. (2014)	90.05
Passos et al. (2014)*	90.90
Luo et al. (2015)* + gaz	89.9
Luo et al. (2015)* + gaz + linking	**91.2**
Chiu and Nichols (2015)	90.69
Chiu and Nichols (2015)*	90.77
LSTM-CRF (no char)	90.20
LSTM-CRF	**90.94**
S-LSTM (no char)	87.96
S-LSTM	90.33

Table 1: English NER results (CoNLL-2003 test set). * indicates models trained with the use of external labeled data

Model	F_1
Florian et al. (2003)*	72.41
Ando and Zhang (2005a)	75.27
Qi et al. (2009)	75.72
Gillick et al. (2015)	72.08
Gillick et al. (2015)*	76.22
LSTM-CRF – no char	75.06
LSTM-CRF	**78.76**
S-LSTM – no char	65.87
S-LSTM	75.66

Table 2: German NER results (CoNLL-2003 test set). * indicates models trained with the use of external labeled data

Model	F_1
Carreras et al. (2002)	77.05
Nothman et al. (2013)	78.6
Gillick et al. (2015)	78.08
Gillick et al. (2015)*	**82.84**
LSTM-CRF – no char	73.14
LSTM-CRF	**81.74**
S-LSTM – no char	69.90
S-LSTM	79.88

Table 3: Dutch NER (CoNLL-2002 test set). * indicates models trained with the use of external labeled data

Model	F_1
Carreras et al. (2002)*	81.39
Santos and Guimarães (2015)	82.21
Gillick et al. (2015)	81.83
Gillick et al. (2015)*	82.95
LSTM-CRF – no char	83.44
LSTM-CRF	**85.75**
S-LSTM – no char	79.46
S-LSTM	83.93

Table 4: Spanish NER (CoNLL-2002 test set). * indicates models trained with the use of external labeled data

IDCNN+CRF

IDCNN+CRF

While previous models are expressive and accurate, they fail to fully exploit the parallelism opportunaties of a GPU.

> Fast and Accurate Entity Recognition with Iterated Dilated Convolutions
>
> ——EMNLP2017

- Problems
  - LSTMs requires $O(N)$ time when performing sequential processing on sentences of length N.
  - CNN's computational cost grows with the number of layers, while its representation is limited by the effective input width.
  - Pooling on sequence is not appropriate for sequence labeling.
- This paper
  - The effective input width can grow exponentially with the depth, with no loss in resolution at each layer and with modest number of parameters.
  - The size of the effective input width for a token at layer l: $l(w-1)+1 \rightarrow 2^{l+1}-1$.

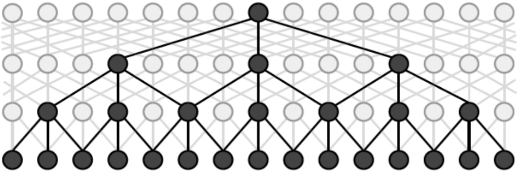
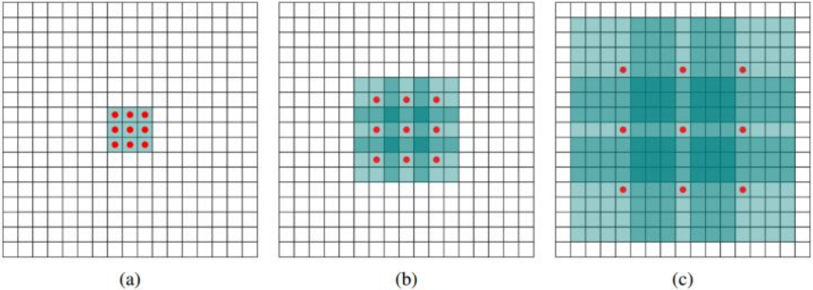


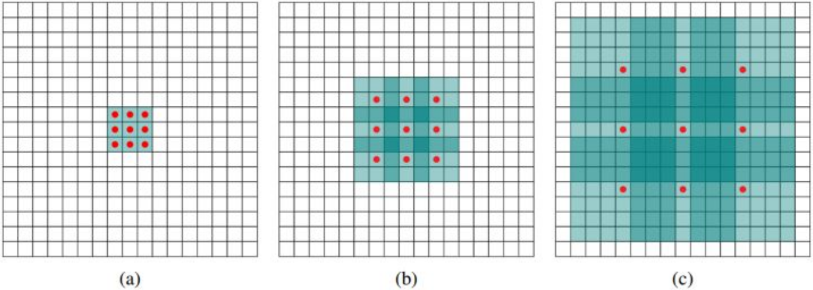
Figure 1: A dilated CNN block with maximum dilation width 4 and filter width 3. Neurons contributing to a single highlighted neuron in the last layer are also highlighted.

3 stacked dilated convolutions of width 3 produces token representations with a effective input width of 15 tokens.

Fast and .... Dilated Convolutions Dilated Convolutions



- (a)1-dilated conv， like normal convolution.
- (b)2-dilated conv， with effective input width of 7.
- (c)4-dilated conv， with effective input width of 15.



- The convolutional operator has changed:

$$c_t = W_c \bigoplus_{k=0}^{r} x_{t \pm k} \rightarrow c_t = W_c \bigoplus_{k=0}^{r} x_{t \pm k\delta}$$

where δ is the dilation width.

input text: $x = [x_1, x_2, \ldots, x_T]$, output tags: $y = [y_1, y_2, \ldots, y_T]$

1. Conditionally independent model:

$$P(y|x) = \prod_{t=1}^{T} P(y_t|F(x))$$

2. Linear-chain CRF model:

$$P(y|x) = \frac{1}{Z_x} \prod_{t=1}^{T} \varphi_t(y_t|F(x))\varphi_p(y_t, y_{t-1})$$

CRF imposes more prior knowledge about the structure of the interactions among the tags, while accompaning worse computational complexity than independent prediction.

- Stacked dilated CNNs can easily incorporate global informatin from a whole sentence or document.
- Simply increasing the depth of stacked dilated CNNs causes considerable OVERFITTING.

This paper presents Iterated Dilated CNNs(ID-CNNs), which instead apply the same small of dilated convolutions multiple times, each itetate taking as input the result of the last application.

Model Architecture:

- IN A BLOCK B

1. The first layer transforms the input to i_t:

$$i_t = D_1^{(0)} x_t$$

2. The stack of layers with the following recurrence:

$$c_t^{(j)} = ReLU(D_{2^{L_C-1}}^{(j-1)} c_t^{(j-1)})$$

3. Add a final dilation-1 layer to the stack:

$$c_t^{(L_C+1)} = ReLU(D_1^{(L_C)} c_t^{(L_C)})$$

- APPLY B L_b TIMES

$$b_t^{(1)} = B(i_t) \qquad b_t^{(k)} = B(b_t^{(k-1)}) \qquad h_t^{(L_b)} = W_o b_t^{(L_b)}$$

Training:

- Tags are conditioanlly independent:

$$\frac{1}{T}\sum_{t=1}^{T}\log P(y_t|h_t^{(L_b)})$$

- To let subsequent blocks learn to correct dependency violarions of their predecessors:

$$\frac{1}{L_b}\sum_{k=1}^{L_b}\frac{1}{T}\sum_{t=1}^{T}\log P(y_t|h_t^{(k)})$$

First invectigation of *dropout with expectation-linear regularization* for NLP (Ma et al.,2017).

Fast and .... Dilated Convolutions Experimental Results

Model	F1
Ratinov and Roth (2009)	86.82
Collobert et al. (2011)	86.96
Lample et al. (2016)	90.33
Bi-LSTM	89.34 ± 0.28
4-layer CNN	89.97 ± 0.20
5-layer CNN	90.23 ± 0.16
ID-CNN	90.32 ± 0.26
Collobert et al. (2011)	88.67
Passos et al. (2014)	90.05
Lample et al. (2016)	90.20
Bi-LSTM-CRF (re-impl)	90.43 ± 0.12
ID-CNN-CRF	$\mathbf{90.54 \pm 0.18}$

Table 1: F1 score of models observing sentence-level context. No models use character embeddings or lexicons. Top models are greedy, bottom models use Viterbi inference .

Model	Speed
Bi-LSTM-CRF	$1\times$
Bi-LSTM	$9.92\times$
ID-CNN-CRF	$1.28\times$
5-layer CNN	$12.38\times$
ID-CNN	$14.10\times$

Table 2: Relative test-time speed of sentence models, using the fastest batch size for each model.[5]

Fast and .... Dilated Convolutions Experimental Results

Model	w/o DR	w/ DR
Bi-LSTM	88.89 ± 0.30	$\mathbf{89.34} \pm 0.28$
4-layer CNN	89.74 ± 0.23	$\mathbf{89.97} \pm 0.20$
5-layer CNN	89.93 ± 0.32	$\mathbf{90.23} \pm 0.16$
Bi-LSTM-CRF	90.01 ± 0.23	$\mathbf{90.43} \pm 0.12$
4-layer ID-CNN	89.65 ± 0.30	$\mathbf{90.32} \pm 0.26$

Table 3: Comparison of models trained with and without expectation-linear dropout regularization (DR). DR improves all models.

Model	F1
4-layer ID-CNN (sent)	90.32 ± 0.26
Bi-LSTM-CRF (sent)	90.43 ± 0.12
4-layer CNN $\times$ 3	90.32 ± 0.32
5-layer CNN $\times$ 3	90.45 ± 0.21
Bi-LSTM	89.09 ± 0.19
Bi-LSTM-CRF	90.60 ± 0.19
ID-CNN	$\mathbf{90.65 \pm 0.15}$

Table 4: F1 score of models trained to predict document-at-a-time. Our greedy ID-CNN model performs as well as the Bi-LSTM-CRF.

CONCLUSION

Conclusion

1. 使用IDCNN-CRF在保证性能的情况下提升训练速度
2. 根据领域制定更细粒度的实体类型
   - Person(人物、职位等)
   - Location(地点)
   - Organization(机构)
   - Works(作品)
   - Date(日期、时间)
   - Style(艺术风格)
   - Faction(流派)
   - Misc(其他类型)