Fast and Accurate Entity Recognition with Iterated Dilated Convolutions

논문발표자료 (2019.2.9)
모두의 연구소 풀잎스쿨 NLP-Bootcamp 2nd.
발표자 : 백병인

한장짜리 요약

- Sequence Labeling(특히 NER) 문제를 푸는 데는 Bi-LSTM-CRF가 (이때까지는) 대세라고 한다.
- 그러나 RNN을 Encoder로 쓰는 것은 GPU 병렬처리 측면에서 비효율적이니 CNN을 쓰고 싶다. 그러나 RNN의 강점인 Long-range dependency를 충분히 고려하는데 약점이 있다.
- 그래서 Dilated CNN을 적용하여 RNN과 CNN의 장점을 둘 다 살리는 방법을 생각해 보았다.
- Bi-LSTM-CRF만큼의 accuracy가 나오면서도 14~20배나 빠른 모델을 구축할 수 있었다.

Named Entity Recognition (NER)

From Wikipedia

Most research on NER systems has been structured as taking an unannotated block of text, such as this one:

Jim bought 300 shares of Acme Corp. in 2006. $x = [x_1, \ldots, x_T]$

And producing an annotated block of text that highlights the names of entities:

[Jim]$_{\text{Person}}$ bought 300 shares of [Acme Corp.]$_{\text{Organization}}$ in [2006]$_{\text{Time}}$. $y = [y_1, \ldots, y_T]$

NER as a discriminative Task

$$x = [x_1, \ldots, x_T] \quad y = [y_1, \ldots, y_T]$$

Yt끼리 독립적으로 추론한다면?

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

Yt끼리의 관련성을 함께 고려한다면? CRF!!

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

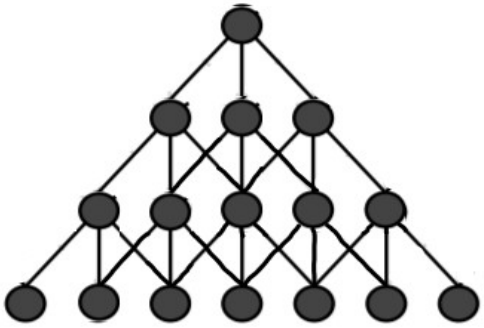
- 사실 이 논문은 x의 인코더 F(x)를 어떻게 구현할지에 대한 것이다.
- CRF가 정답이다만, 너무 느린 것은 어쩔 수 없다.
- F(x) 안에서 간접적으로라도 이웃간 관련성이 고려될 수 있다면 좋겠다.
- 그래서 ID-CNN??

CNN & Dilated CNN

- l : layer 개수 (l =3)
- W : convolution width (w=3)
- r : l번째 layer가 참조하는 토큰 개수

CNN

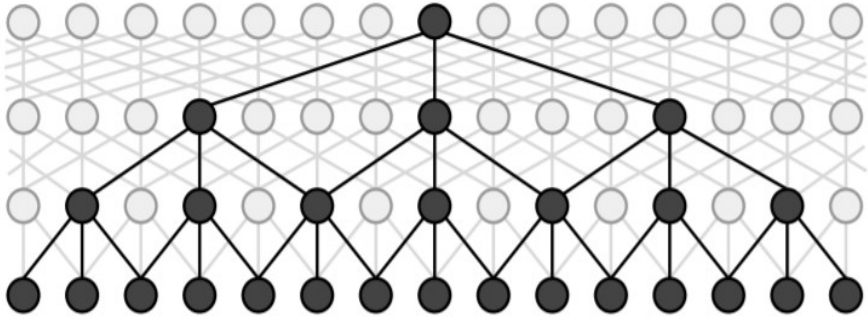
$$c_t = W_c \bigoplus_{k=0}^{r} x_{t \pm k}$$



- r = l(w − 1) + 1 = 3*(3-1)+1 = 7
- r이 l에 선형적 비례

Dilated CNN

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



- r = (w − 1)^(l+1) – 1 = (3-1)^(3+1)-1=15
- r이 l에 따라 지수적으로 증가

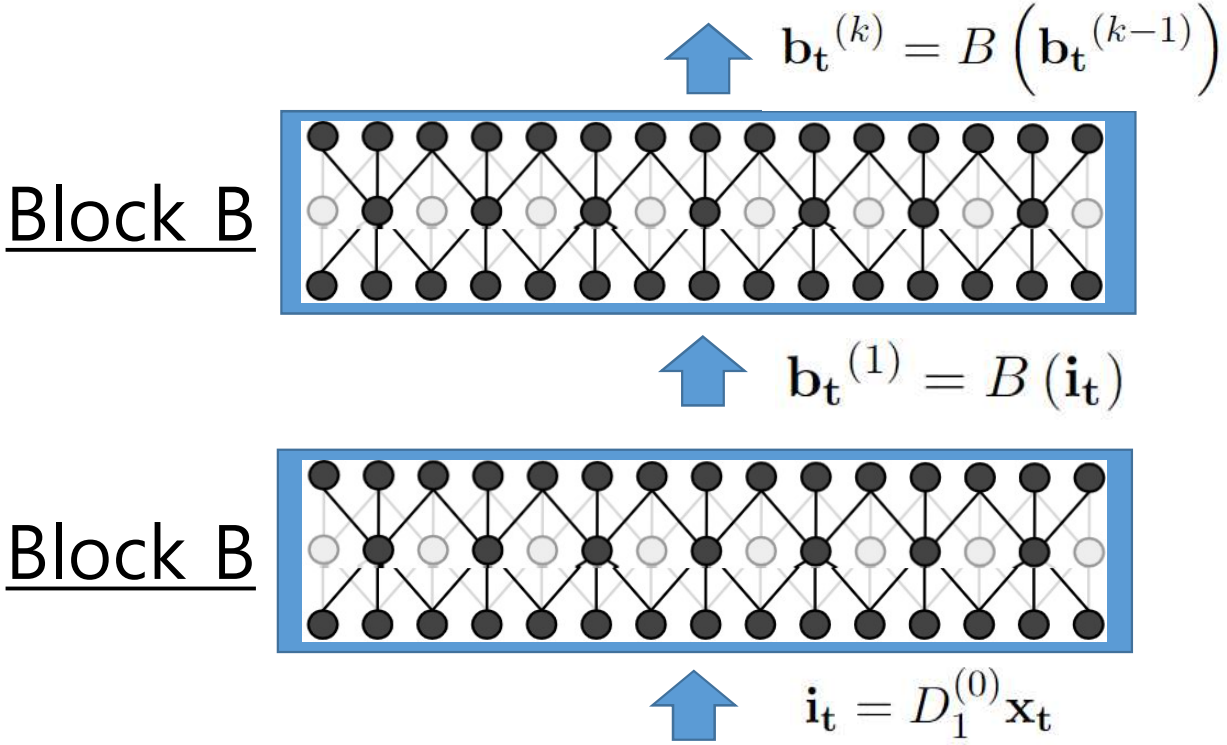
- Dilated CNN을 쓰면 기존 CNN보다 적은 layer를 통해서 훨씬 긴 범위의 토큰 입력을 인코딩 가능하다.

그런데 무슨 문제라도??

- Dilated CNN을 쓰면,
  - 4-Layer만으로도 31token 커버 가능.(PTB 평균 문장 23token)
  - 8-Layer로 1000token 이상 (신문기사 커버 가능)
- 그런데… Overfitting이 발생한다.
  - Dilated CNN layer를 깊이 쌓으면 트레이닝이 안된다.
- 그래서 (부득이하게) ID-CNN을 제안한다.

Iterated Dilated CNN (ID-CNN)

- Layer를 마냥 쌓을 수 없다면?
- Layer를 몇 개만 쌓아 block을 만든 후 이를 반복해서 사용하면?



Final representation to obtain score

$$\mathbf{h_t}^{(L_b)} = W_o\mathbf{b_t}^{(L_b)}$$

Training

- Let's maximize log-likelihood

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

- Average Loss Minimization

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

장점들

- 계산적으로 훨씬 간단하다.
- CRF처럼 label연관성을 직접고려하지 않더라도, 간접적으로 오류 수정 효과를 기대할 수 있다.
- Vanishing gradient에 더 안전하다.

Dropout

- Dilated CNN의 overfitting 문제의 대안으로 Dropout.
- 그러나, 일반적인 Dropout(Srivastava et. al., 2014)은 DCNN의 test 성능을 떨어뜨린다.
- 그래서 Ma et al.,2017 (dropout with expectation-linear regularization)을 사용하였다.

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	**90.54** ± **0.18**

Model	Speed
Bi-LSTM-CRF	1×
Bi-LSTM	9.92×
ID-CNN-CRF	1.28×
5-layer CNN	12.38×
ID-CNN	14.10×

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