DL Seminar

Deep Learning Paper Review

한양대학교 AI Lab
이 주 홍

한양대학교

Introduction

- Recurrent Convolutional Neural Network for Text Classification
- AAAI 2015에서 발표된 논문
- 제목만 봐도 대충 무슨 내용인지 알 듯한……...
  - Recurrent Convolutional Neural Network 를 이용해서
  - Text Classification 이라는 문제를 해결해보겠다!

Text Classification

- Text Classification 문제를 풀고자 함

- Text Classification이란?
  - 제일 베이직한 NLP task 중 하나
  - Input으로 text (sentence, paragraph)를 받아서, Output으로 Classification 결과를 내보냄
  - 주제 분류 (Topic Classification), 감정 분석 (Sentiment Analysis) 등이 이에 해당됨

Related Work

- Traditional Method
  - Feature Engineering
  - Feature Selection
  - Machine Learning Model

- Neural Method
  - Convolutional Neural Network
  - Recurrent Neural Network

Traditional Method - Feature Engineering

- 좋은 Feature를 만들어보자!

- 좋은 Feature로는 무엇이 있을까요?
  - Bag-of-words
  - N-gram
  - NLP toolkit

Traditional Method - Feature Engineering

- Bag-of-words
  - A: I will play the game after seminar
  - B: Let's play the soccer after seminar

Sentence	i	play	seminar	let's	will	the	after	game	soccer
A	1	1	1	0	1	1	1	1	0
B	0	1	1	1	0	1	1	0	1

자연어 문장 → Feature Vector

변환시킨 vector를 model의 input으로 넣는다

Traditional Method - Feature Engineering

- N-gram
  - A: I will play the game after seminar
  - B: Let's play the soccer after seminar

Sentence	i	play	…	i will	will play	play the	…	soccer after	after seminar
A	1	1	…	1	1	1	…	0	1
B	0	1	…	0	0	1	…	1	1

Traditional Method - Feature Engineering

- N-gram
  - *A sunset stroll along the South **Bank** affords an array of stunning vantage points.*
  - 위 문장에서 ***Bank*** 의 뜻이 무엇일까?
    - 1-gram (uni-gram) : 은행, 둑
    - 2-gram (bi-gram) : 남쪽 은행, 둑...?
    - 5-gram (five-gram) : 런던의 지방 이름 South Bank

Sentence	a	sunset	...	South Bank	Bank affords	...	Stroll along the South Bank	...
A	1	1	...	1	1	...	1	...

N을 키우면 문맥 정보를 보다 많이 얻을 수 있지만
Data Sparsity 문제가 생김

Traditional Method - Feature Engineering

- NLP Toolkit
  - POS Parser
  - Dependency Parser
  - NER tagger
- 과거 연구에는 필수적인 Feature로 사용되었음
- 하지만 NLP Toolkit 자체가 많은 에러를 포함하고 있고,
  이를 ML model에 넣고 학습하면 에러가 더욱 증폭됨
- 최근에는 이런 NLP Toolkit에 의한 Feature 없이,
  순수 raw sentence만을 가지고 model을 학습시키고자 함

Traditional Method - Feature Selection & ML Model

- Feature Selection
  - Feature들 중 쓸모 있는 것들을 추려내는 작업
  - ex) stop-word

- Machine Learning Model
  - Logistic Regression
  - Naïve Bayes Classifier
  - Support Vector Machine

Neural Method

Introduction
Related Work
Model
Experiments
Conclusion

- Convolutional Neural Network

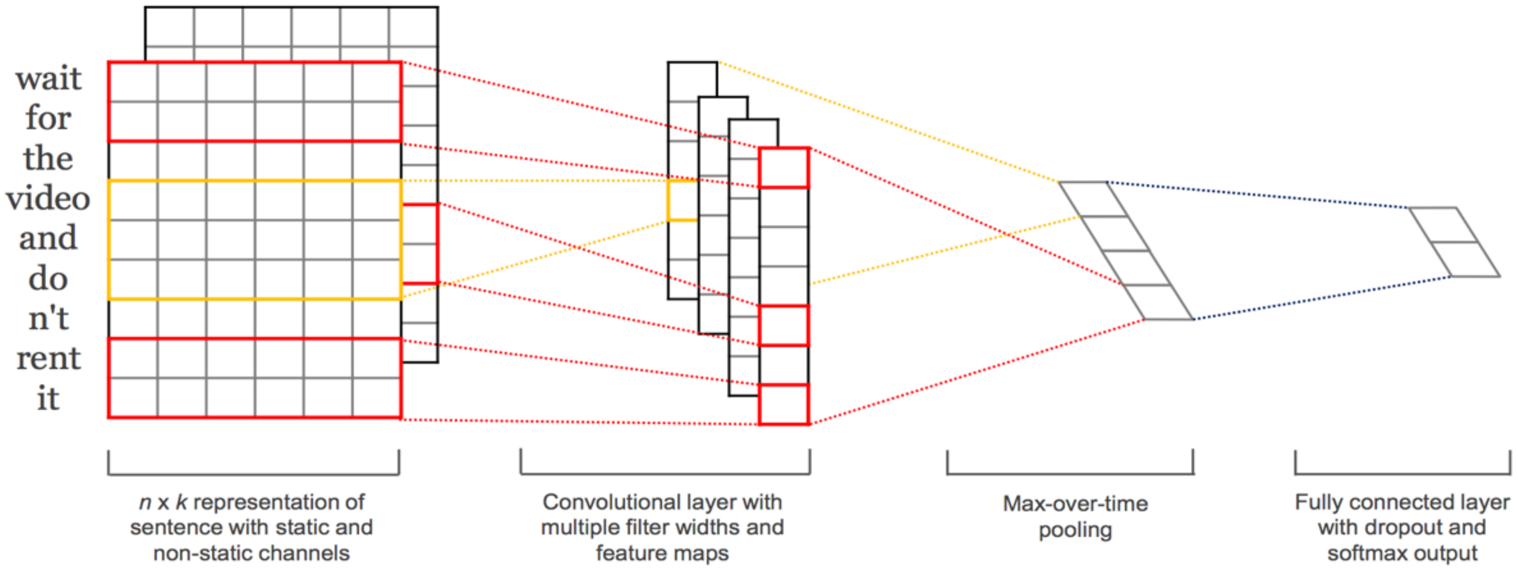


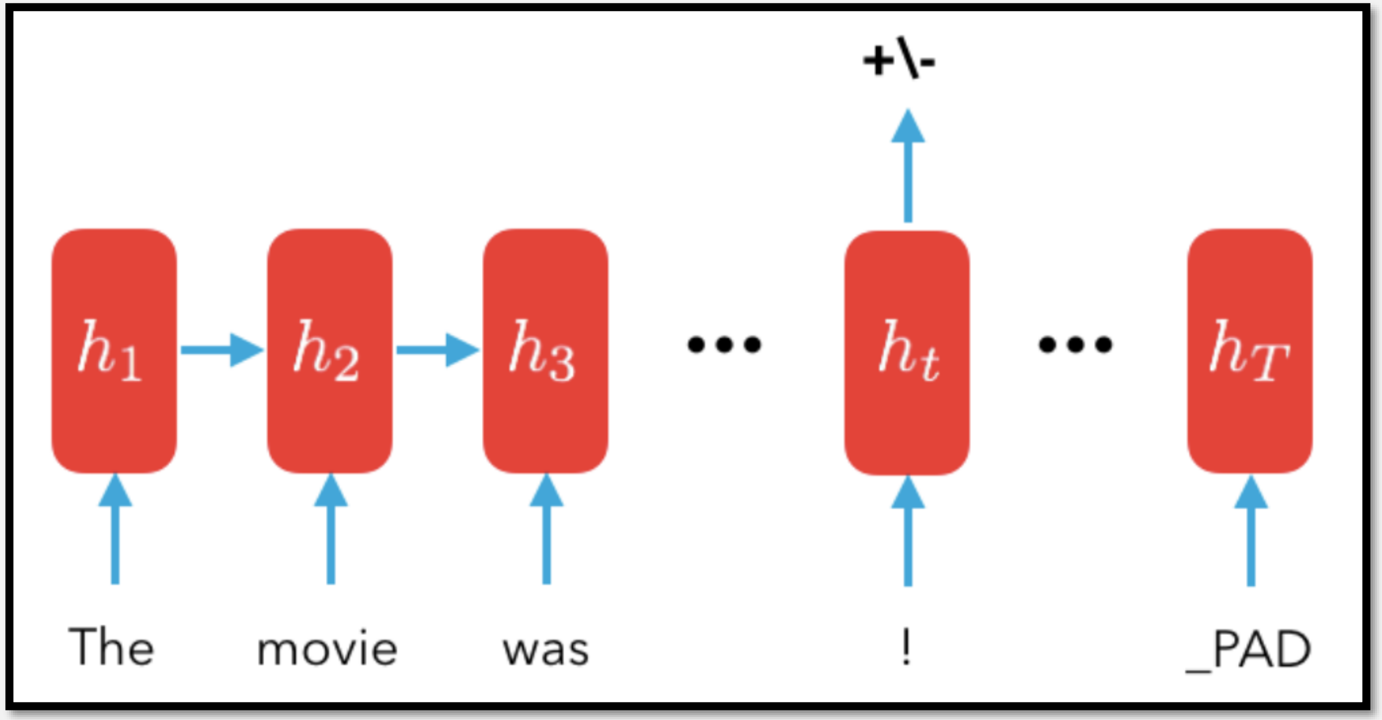
Figure 1: Model architecture with two channels for an example sentence.

한양대학교

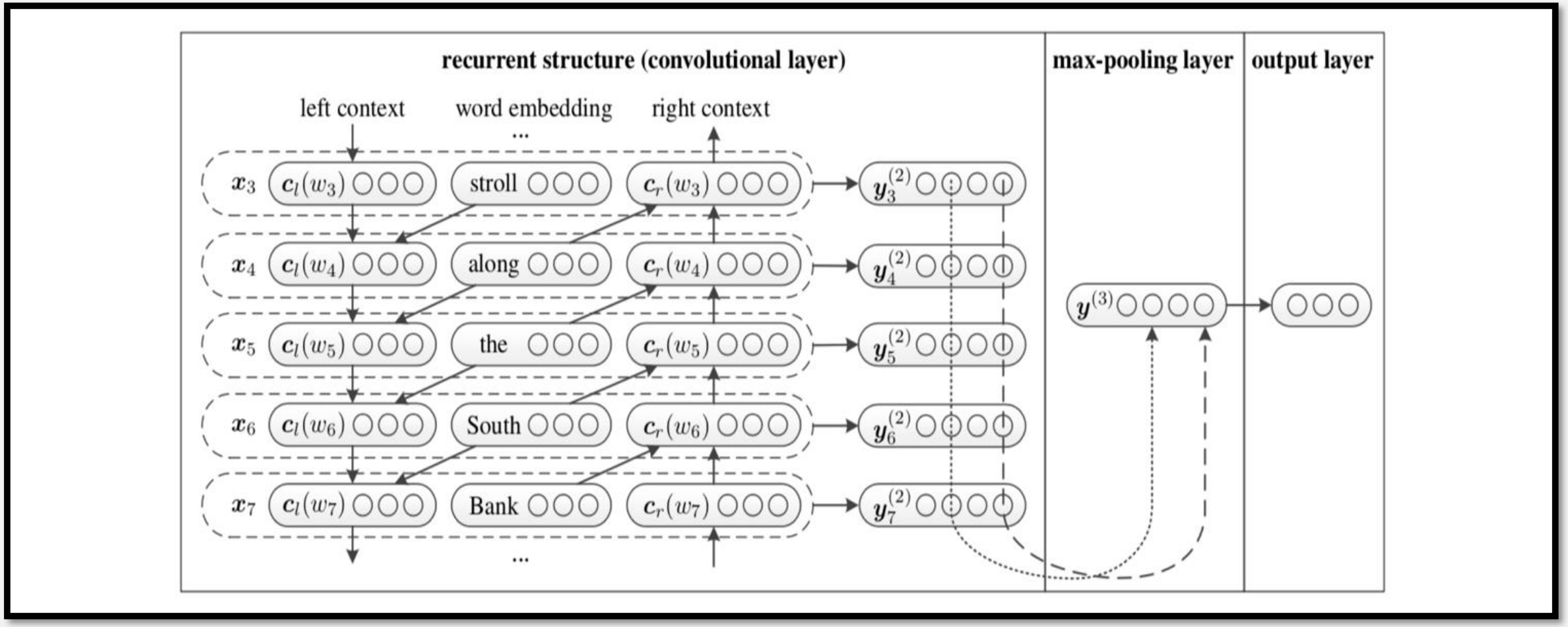
- Introduction
- Related Work
- Model
- Experiments
- Conclusion

Neural Method

- Recurrent Neural Network

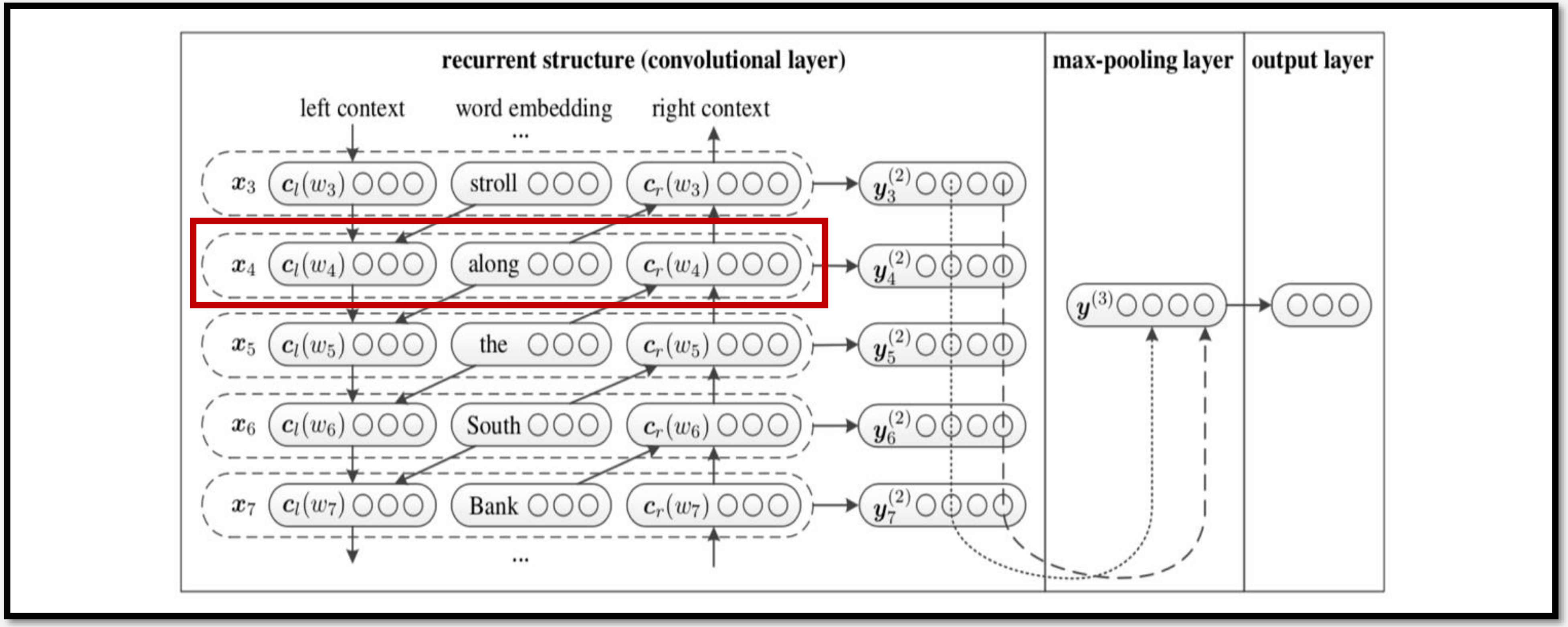


Recurrent Convolutional Neural Network



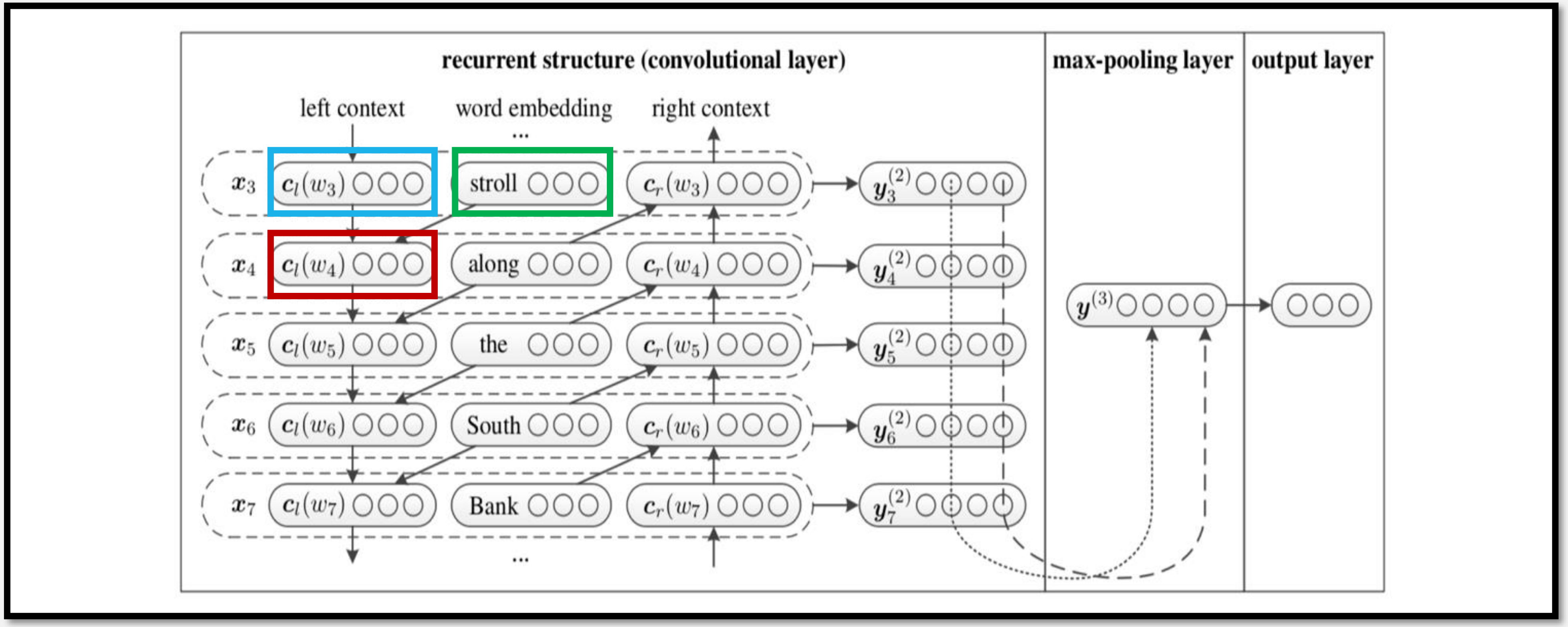
- Recurrent Structure 를 이용하여
- Context Embeddings 을 하고
- 이를 이용해 Convolution 을 모사했더니, 기존 CNN 보다 좋았다!

Recurrent Convolutional Neural Network



- Word Representation: 하나의 단어는 x_i 로 표현함
  - $x_i = [\, c_l(w_i);\ e(w_i);\ c_r(w_i)\,]$
  - $c_l(\cdot)$ *and* $c_r(\cdot)$ *is context embeddings,*
  - $e(\cdot)$ *is word embeddings*

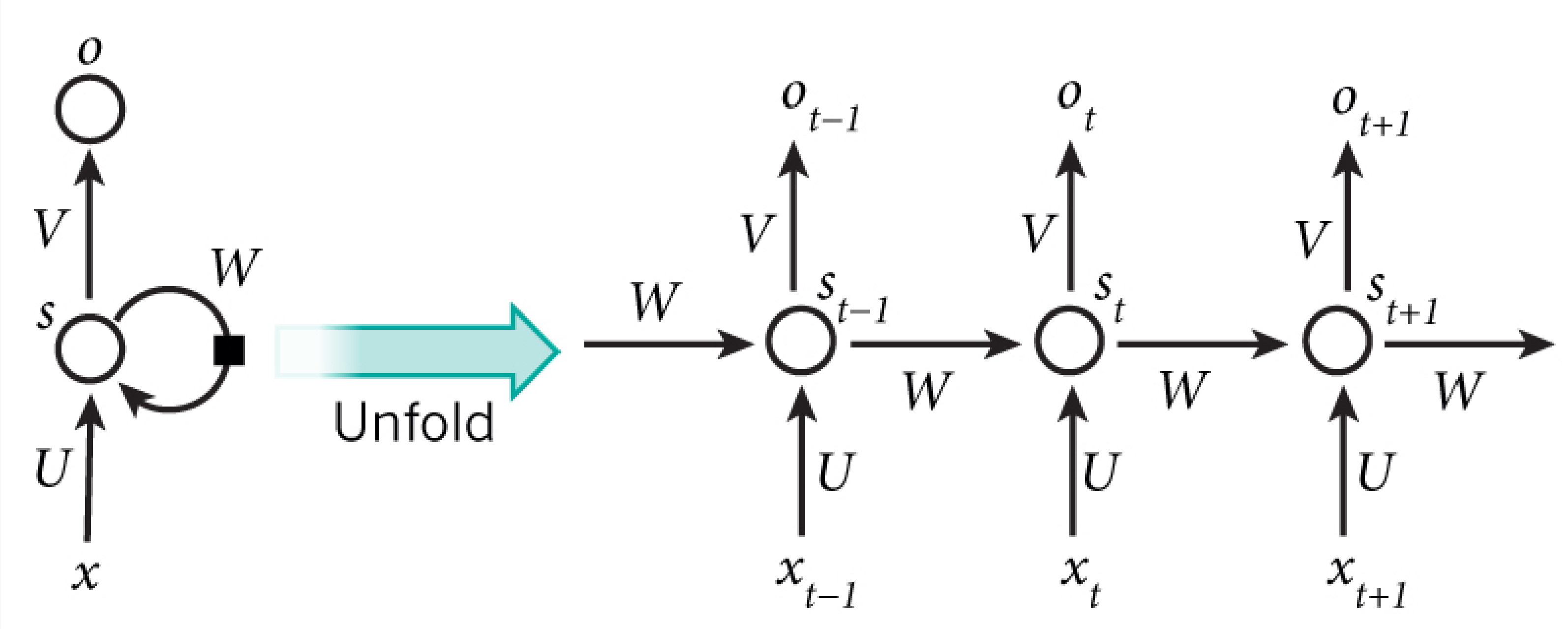
Recurrent Convolutional Neural Network



- Context Embeddings

$$\boxed{c_l(w_i)} = f(W^{(l)}\boxed{c_l(w_{i-1})} + W^{(sl)}\boxed{e(w_{i-1})})$$

Recurrent Convolutional Neural Network

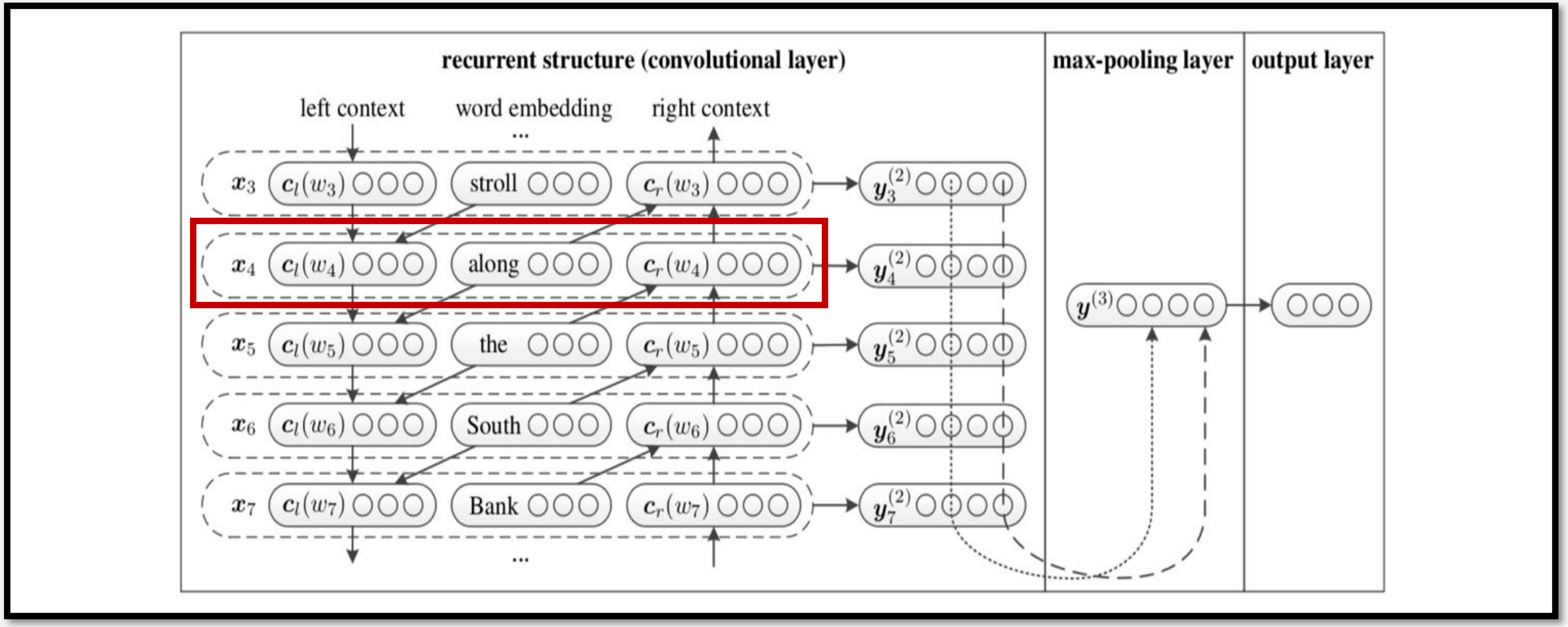


$$\boxed{s_t} = f(U\boxed{x_t} + W\boxed{s_{t-1}})$$

- Context Embeddings

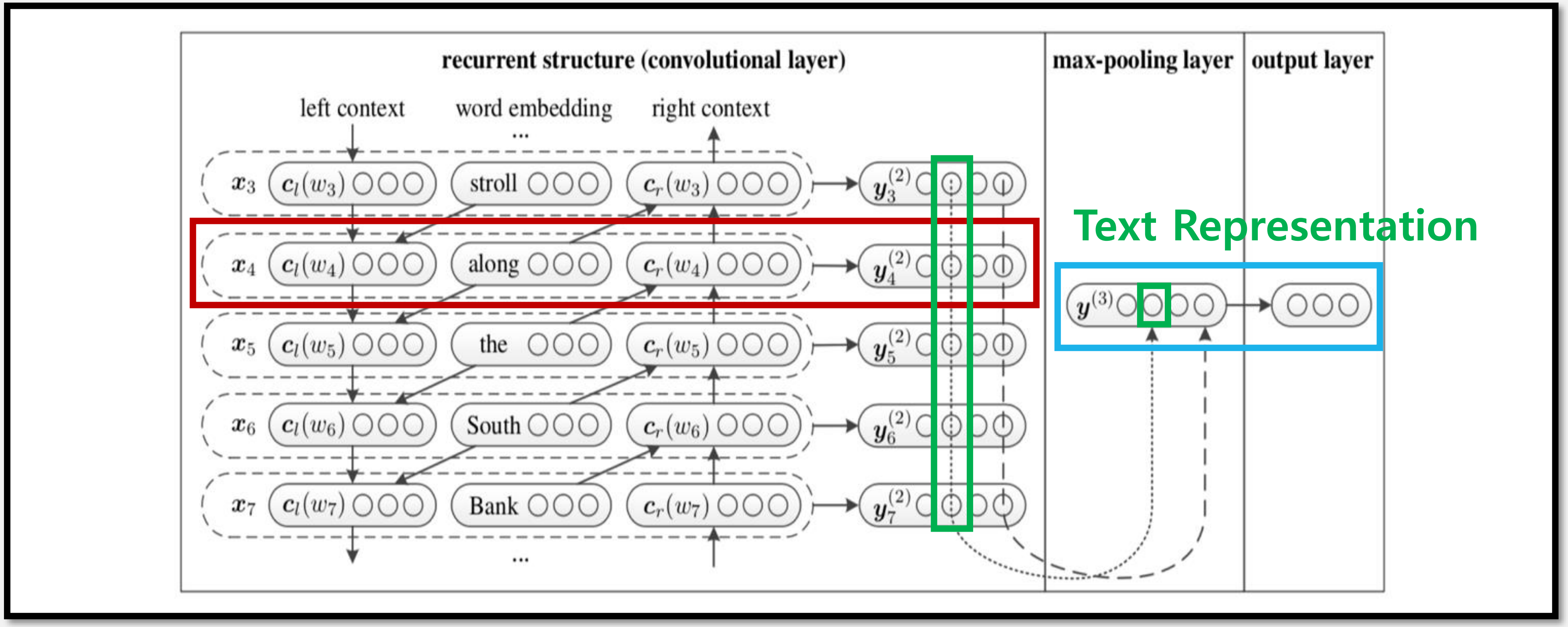
$$\boxed{\mathbf{c}_l(w_i)} = f(W^{(l)}\boxed{\mathbf{c}_l(w_{i-1})} + W^{(sl)}\boxed{\mathbf{e}(w_{i-1})})$$

Recurrent Convolutional Neural Network



- Word Representation: 하나의 단어는 x_i 로 표현함
  - $x_i = [\, c_l(w_i);\ e(w_i);\ c_r(w_i)\,]$
  - $c_l(\cdot)$ *and* $c_r(\cdot)$ *is context embeddings,*
  - $e(\cdot)$ *is word embeddings*

Recurrent Convolutional Neural Network



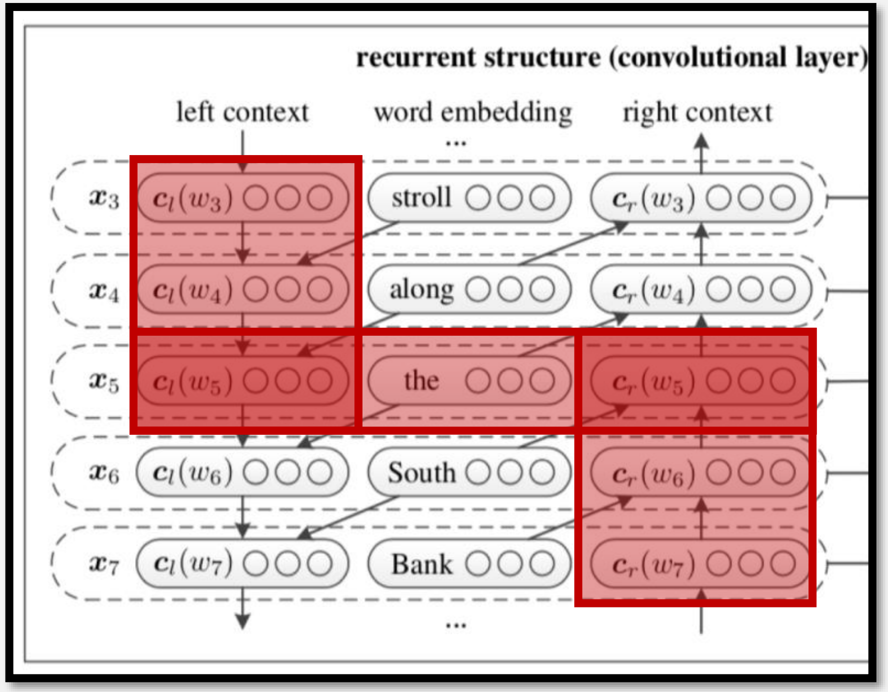
$$\boldsymbol{y}_i^{(2)} = \tanh\left(W^{(2)}\boldsymbol{x}_i + \boldsymbol{b}^{(2)}\right) \tag{4}$$

$$\boldsymbol{y}^{(3)} = \max_{i=1}^{n} \boldsymbol{y}_i^{(2)} \tag{5}$$

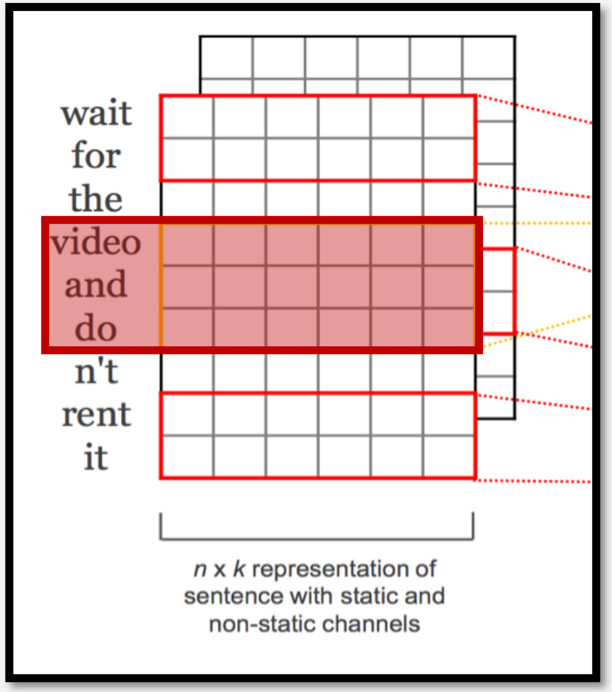
$$\boldsymbol{y}^{(4)} = W^{(4)}\boldsymbol{y}^{(3)} + \boldsymbol{b}^{(4)} \tag{6}$$

- 이후 아키텍처는 매우 심플함

Comparison with CNN



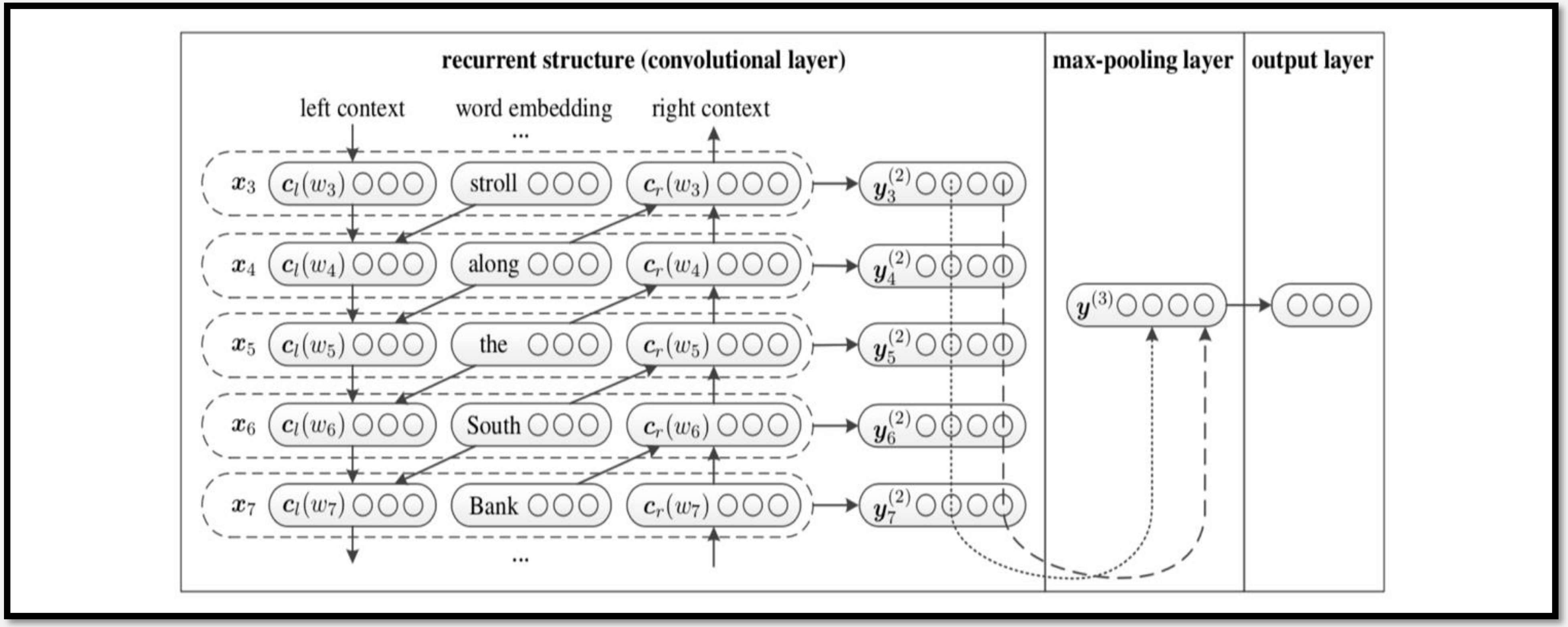
Recurrent Convolutional NN



Convolutional NN

- CNN은 window size 만큼의 context 정보만 사용
- RCNN은 **앞 뒤 전체의 context** 정보를 사용

Recurrent Convolutional Neural Network



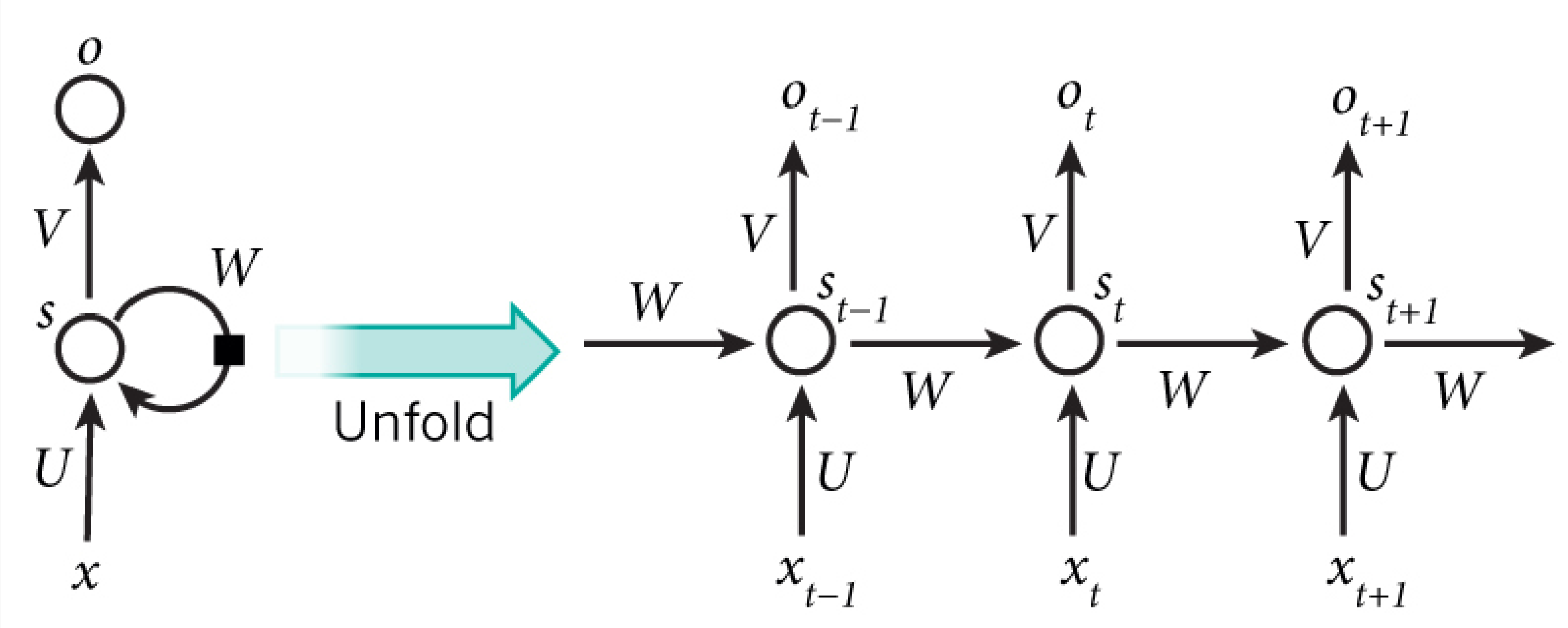
- 결론적으로,
- CNN도 아닌, RNN도 아닌 혼종...
- RNN 구조를 이용해, 기존 CNN의 window based Convolution보다 문맥 정보를 잘 활용하는 모델을 제안 ~~(근데 이름을 이렇게 짓다니...)~~

Experiments

Model	20News	Fudan	ACL	SST
BoW + LR	92.81	92.08	46.67	40.86
Bigram + LR	93.12	92.97	47.00	36.24
BoW + SVM	92.43	93.02	45.24	40.70
Bigram + SVM	92.32	93.03	46.14	36.61
Average Embedding	89.39	86.89	41.32	32.70
ClassifyLDA-EM (Hingmire et al. 2013)	93.60	-	-	-
Labeled-LDA (Li, Sun, and Zhang 2008)	-	90.80	-	-
CFG (Post and Bergsma 2013)	-	-	39.20	-
C&J (Post and Bergsma 2013)	-	-	**49.20**	-
RecursiveNN (Socher et al. 2011b)	-	-	-	43.20
RNTN (Socher et al. 2013)	-	-	-	45.70
Paragraph-Vector (Le and Mikolov 2014)	-	-	-	**48.70**
CNN	94.79	94.04	47.47	46.35
RCNN	**96.49**	**95.20**	**49.19**	47.21

- 당연한 얘기지만, 다른 모델보다 성능이 잘 나온다고 함

Implementation

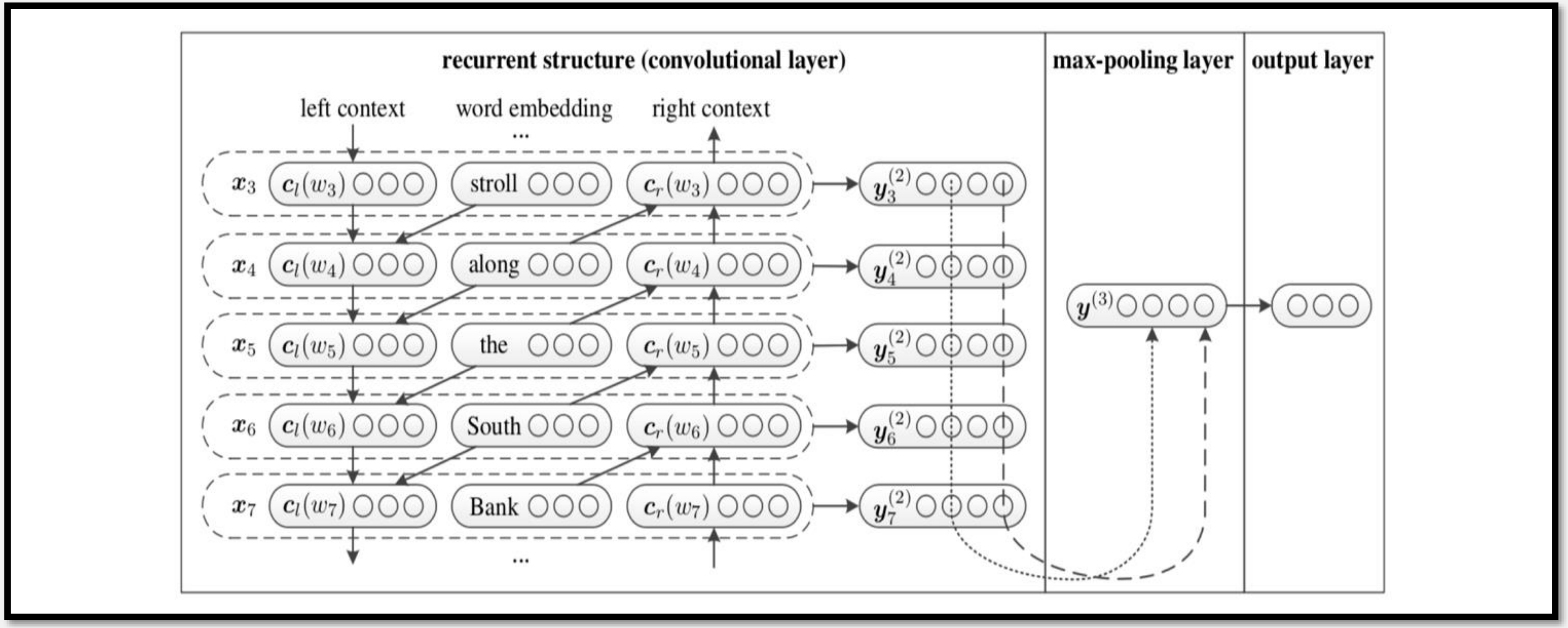


$$\boxed{s_t} = f(U\boxed{x_t} + W\boxed{s_{t-1}})$$

- Context Embeddings

$$\boxed{\mathbf{c}_l(w_i)} = f(W^{(l)}\boxed{\mathbf{c}_l(w_{i-1})} + W^{(sl)}\boxed{\mathbf{e}(w_{i-1})})$$

Implementation

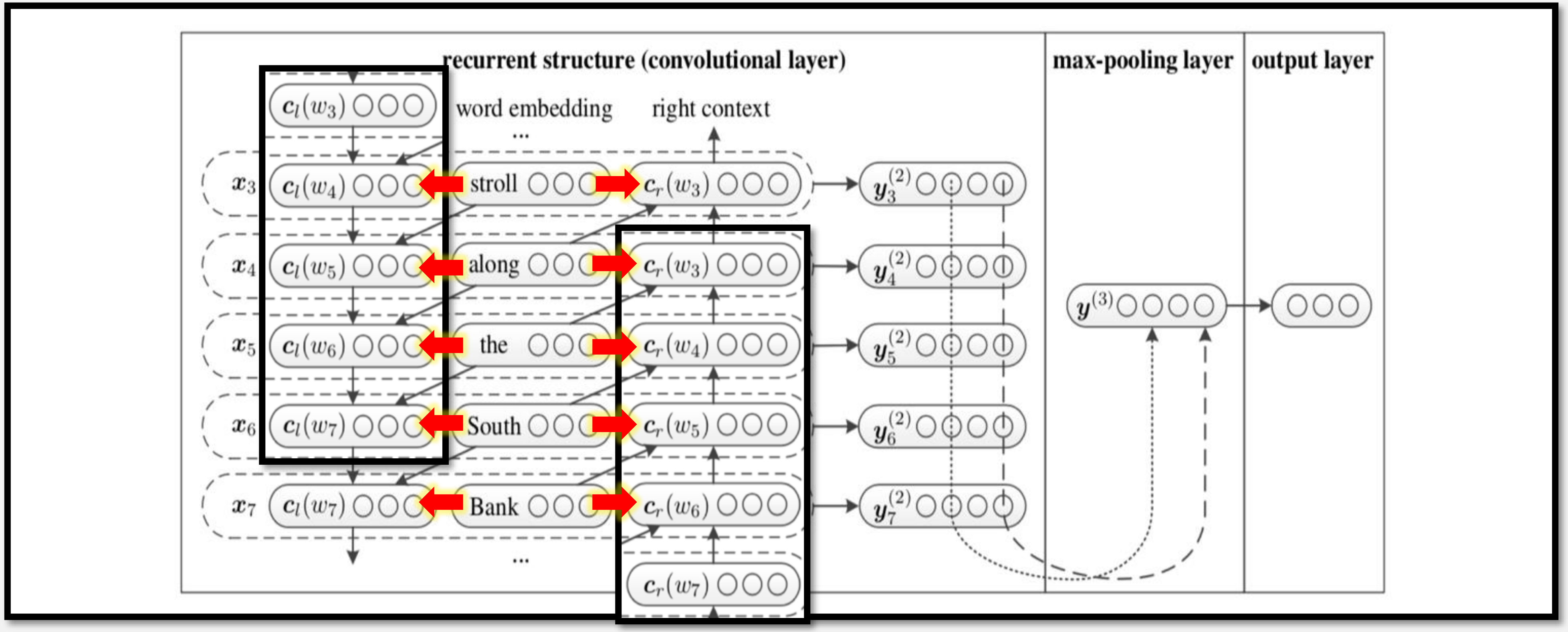


- 구현해보려고 했으나, 당최 어떻게 해야 할 지 감이 오지 않음
- 특히, recurrent structure...

한양대학교

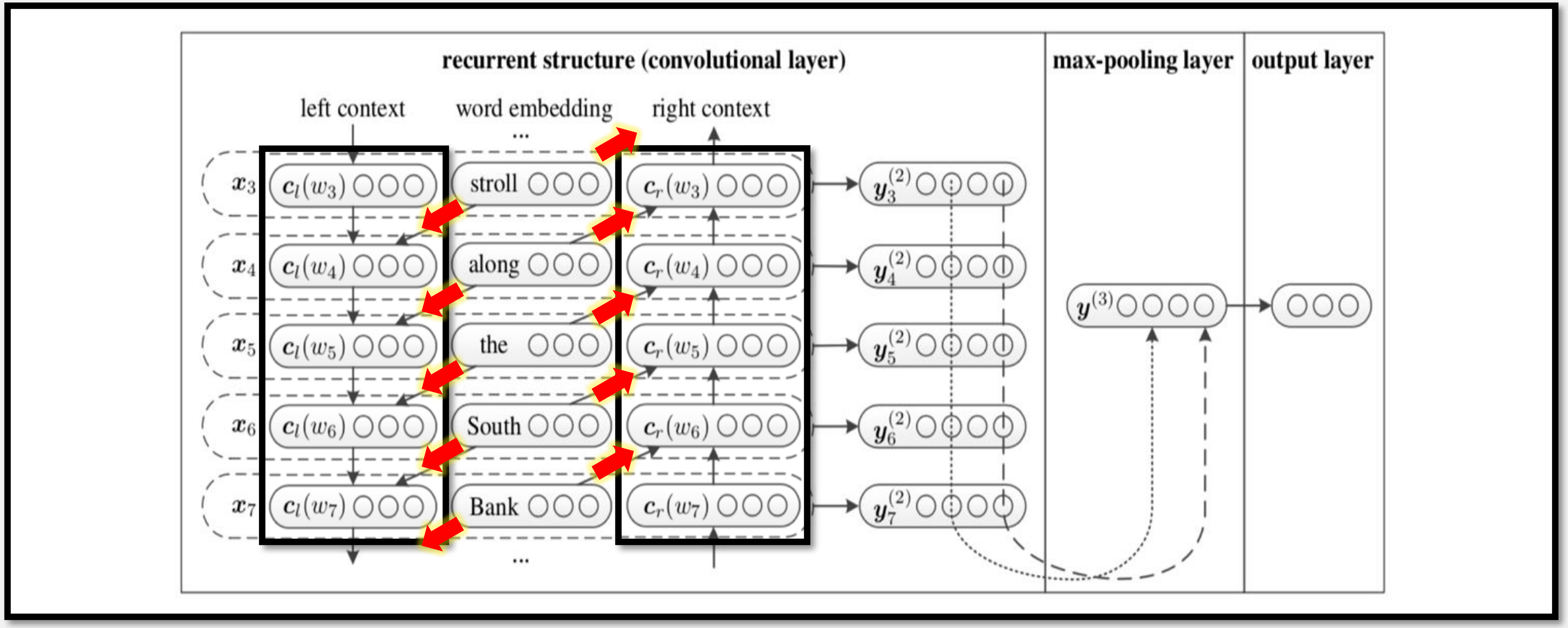
- Introduction
- Related Work
- Model
- Experiments
- Conclusion

Implementation



- 일반 RNN을 위와 같이 구성하여 output (state)를 취함

Implementation



- Output과 word embedding을 합칠 때,
  output을 한 칸(cell)씩 밀어서 concatenate하면 완성!

Discussion

P	well *worth* the; a *wonderful* movie; even *stinging* at; and *invigorating* film; and *ingenious* entertainment; and *enjoy* .; 's *sweetest* movie
N	A *dreadful* live-action; Extremely *boring* .; is *n't* a; 's *painful* .; Extremely *dumb* .; an *awfully* derivative; 's *weaker* than; incredibly *dull* .; very *bad* sign;

- Max-pooling 할 때, 선택되는 단어를 뽑아봄
- Classification 하는데 큰 영향을 준 단어라고 보면 될 듯
- 나름 그럴싸한 단어를 잘 찾아내는 것으로 보임

Conclusion & Future Work

- Conclusion (Contribution)
  - CNN보다 문맥 정보를 잘 활용하는 RNN 기반의 Network을 제안
  - CNN이 문맥정보를 잘 활용하지 못한다는 것을 보여줌
- Future Work
  - Recurrent structure로 convolution을 모사한 부분 이외에는 너무나도 심플하기 때문에 뒷 구조를 보강하면 성능을 높일 수 있지 싶음
  - 저자가 제안한 Network을 기반으로 다양한 NLP task에 적용
  - 지금 하고 있는 Relation Extraction 에도 적용시켜보려 함

Thank You

made by. 이 주 홍