

# Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths

이봉석

- **Author**
  - Yan Xu
  - Lili Mou
  - Ge Li
  - Yunchuan Chen
  - Hao Peng
  - Zhi Jin†
- **Title of Conference(Journal)**
  - EMNLP 2015

# 01. Introduction

- **Relation classification**

- Pairs of 엔티티들에서 의미 있는 relation들을 찾는 task
- 이 task는 information extraction, question answering같은 NLP applications에 유용하다.

Ex) A trillion gallons of <e1> **water** </e1> have been poured into an empty <e2> **region** </e2> of outer space.

⇒ Entity-Destination relation

# 01. Introduction

- 문장에서의 2개의 entity들의 shortest dependency path(SDP)를 사용하였다.
- 또한 LSTM 기반의 RNN을 사용하였다.

# 01. Introduction

- 문장에서의 2개의 entity들의 shortest dependency path(SDP)를 사용하였다.
- 두개의 엔티티의 관계를 결정하기 위해서는 대부분의 관련 정보에 집중하면서 관련 없는 것에서의 noise를 줄이는 것이 좋다고 생각하였다.
- 즉, 오른쪽 예시에서 a, trillion, outer, space은 정보가 적고 노이즈를 가져올 수도 있다.

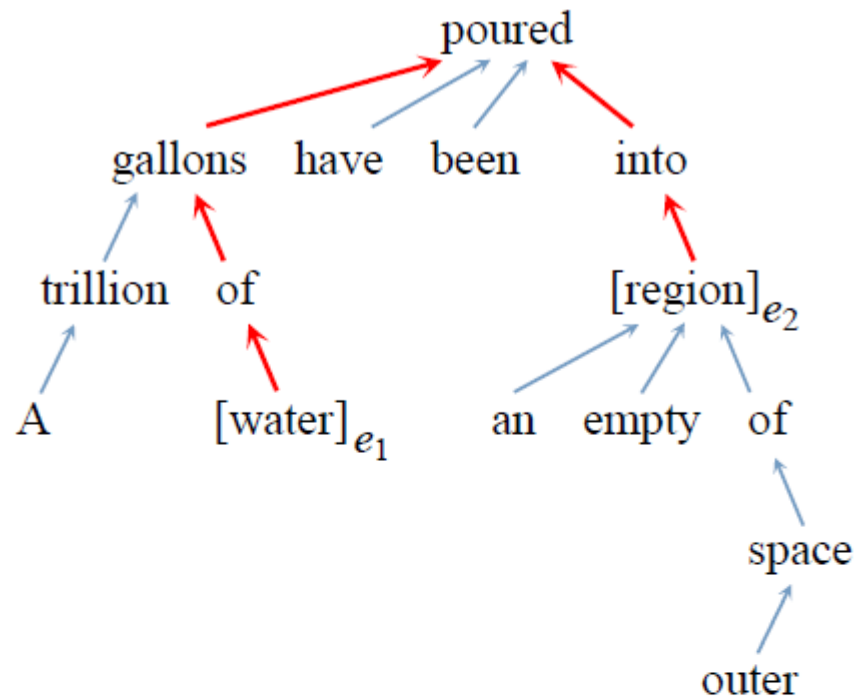


Figure 1: The dependency parse tree corresponding to the sentence “A trillion gallons of water have been poured into an empty region of outer space.” Red lines indicate the shortest dependency path between entities *water* and *region*. An edge  $a \rightarrow b$  refers to  $a$  being governed by  $b$ . Dependency types are labeled by the parser, but not presented in the figure for clarity.

# 01. Introduction

- $r(a, b)$ 와  $r(b, a)$ 은 다르므로 이런 방향을 문제를 해결하기 위해서 SDP를 각각의 엔티티에서 공통조상까지의 2개의 sub-paths로 나눴다.
- 이렇게 추출된 2개의 sub-paths를 classification하기 위해서 concat한다.
- 또한 상위어 하위어에 대한 지식을 이용했다. 예를 들어 water is a kind of substance 에서 water과 substance는 Communication-Topic 관계이기 보다는 Entity-Destination관계임을 알 수 있다. 그래서 이런 정보도 학습에 사용하기 위해서 다중 채널 LSTM을 사용하여 학습한다.

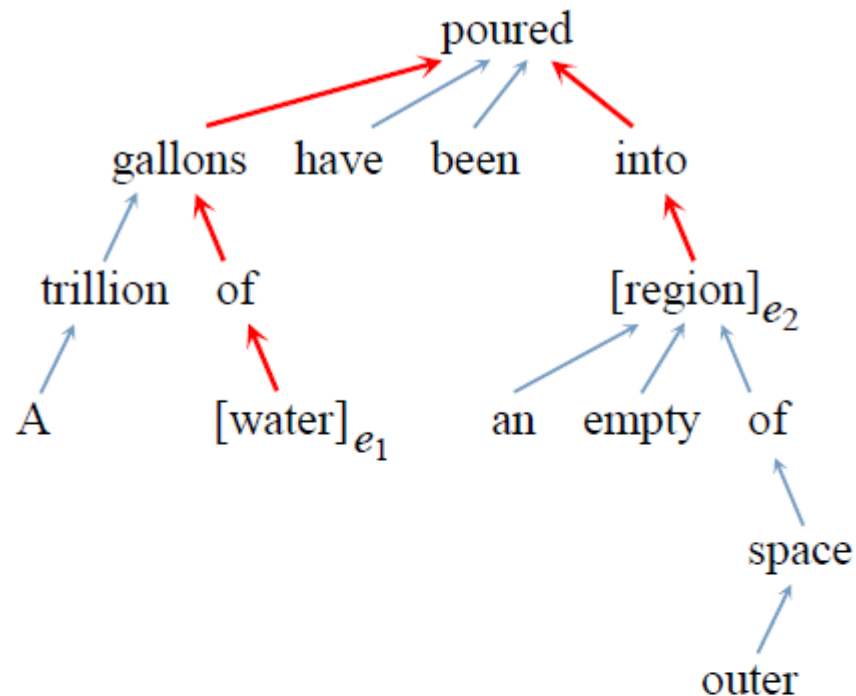


Figure 1: The dependency parse tree corresponding to the sentence “A trillion gallons of water have been poured into an empty region of outer space.” Red lines indicate the shortest dependency path between entities *water* and *region*. An edge  $a \rightarrow b$  refers to  $a$  being governed by  $b$ . Dependency types are labeled by the parser, but not presented in the figure for clarity.

## 02. Model

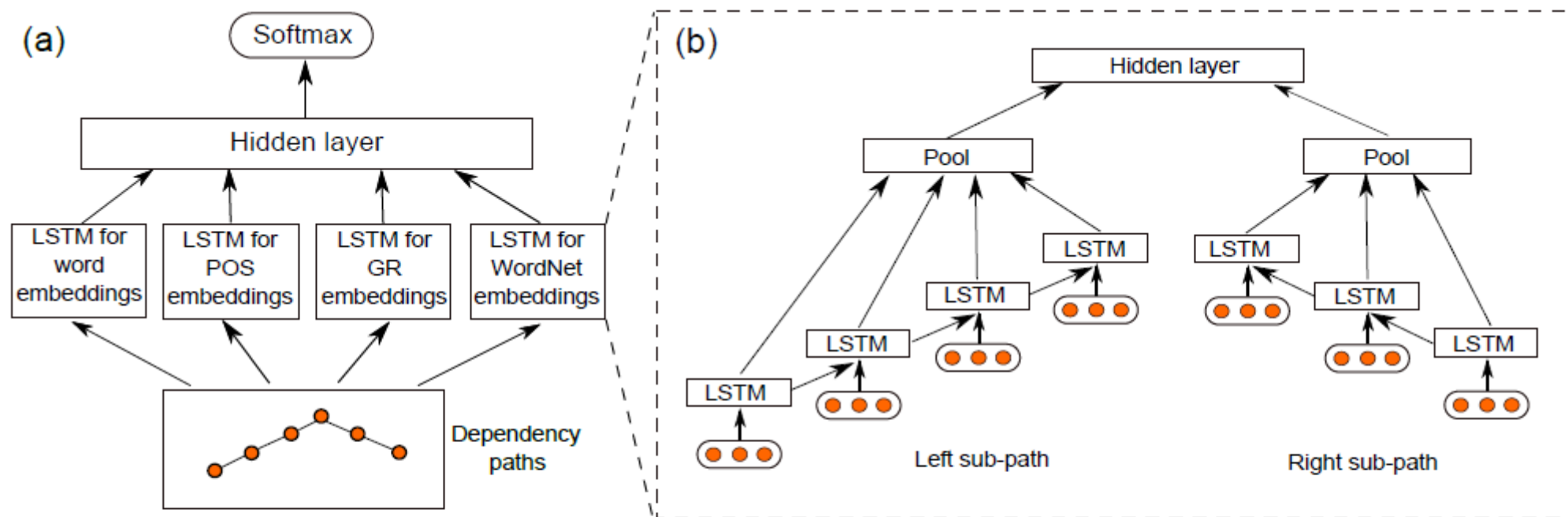
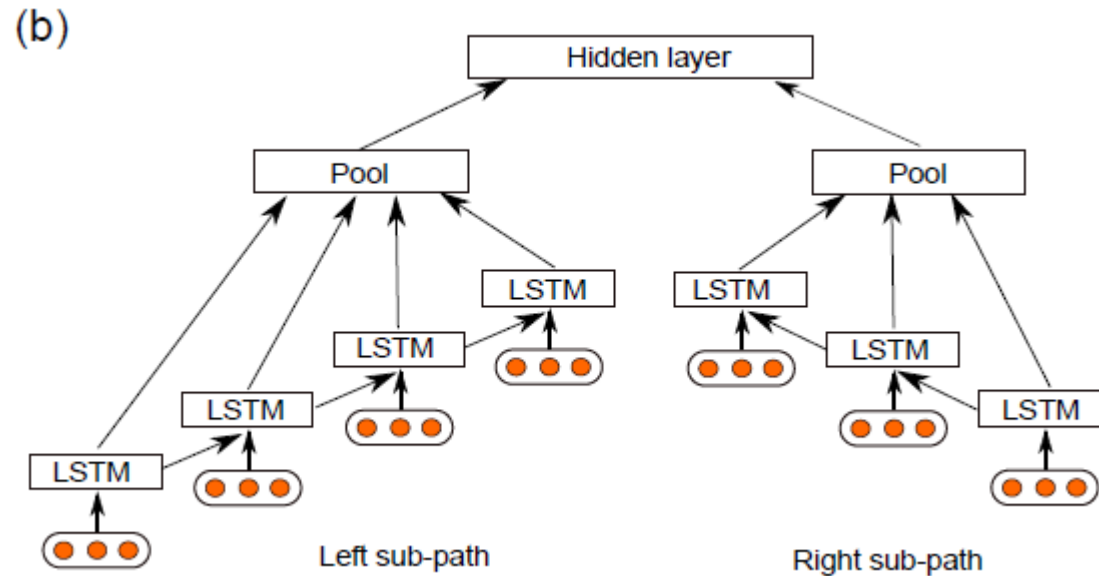


Figure 2: (a) The overall architecture of SDP-LSTM. (b) One channel of the recurrent neural networks built upon the shortest dependency path. The channels are words, part-of-speech (POS) tags, grammatical relations (abbreviated as *GR* in the figure), and WordNet hypernyms.

## 02. Model

- **SDP**

- "Grounded compositional semantics for finding and describing images with sentences" (ACL 2014) 논문을 보면 dependency parse tree는 문장에서의 action과 agents에 focus하기 때문에 자연스럽게 relation classification에 적합하다.



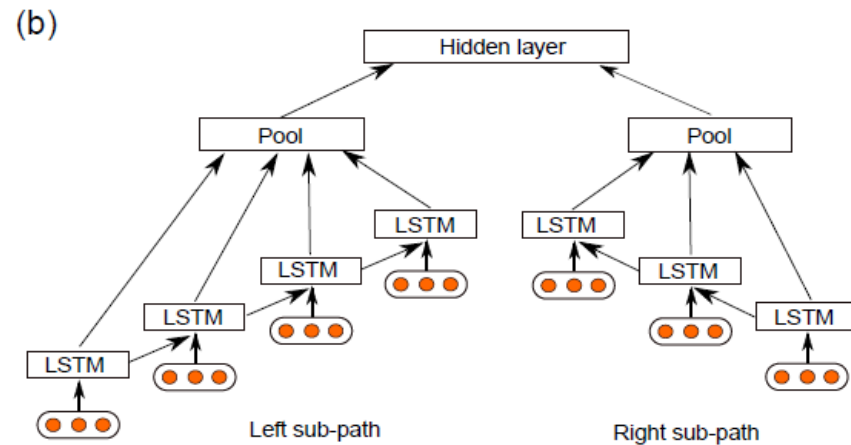
## 02. Model

- SDP

- 아까의 예시를 보면

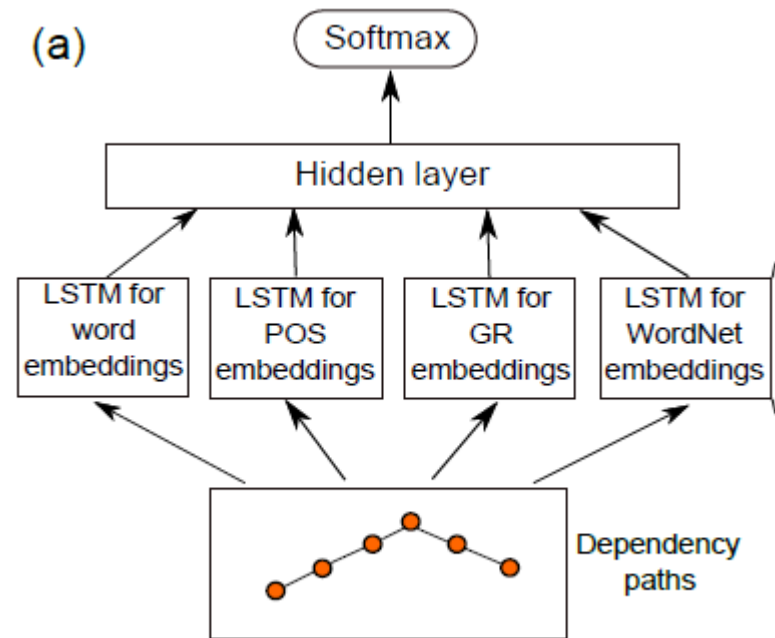
water -> of -> gallons -> poured  
porued <- into <- region

이렇게 2개의 sub-path로 나뉘는데 처음엔 e1의 정보를 capture 하고 다음꺼는 e2의 정보를 capture한다. 이렇게 하므로써 이 문장에서 Entity-Destination(e2, e1)보다는 Entity-Destination(e1, e2)와 관련이 있다는 것을 알 수 있다.





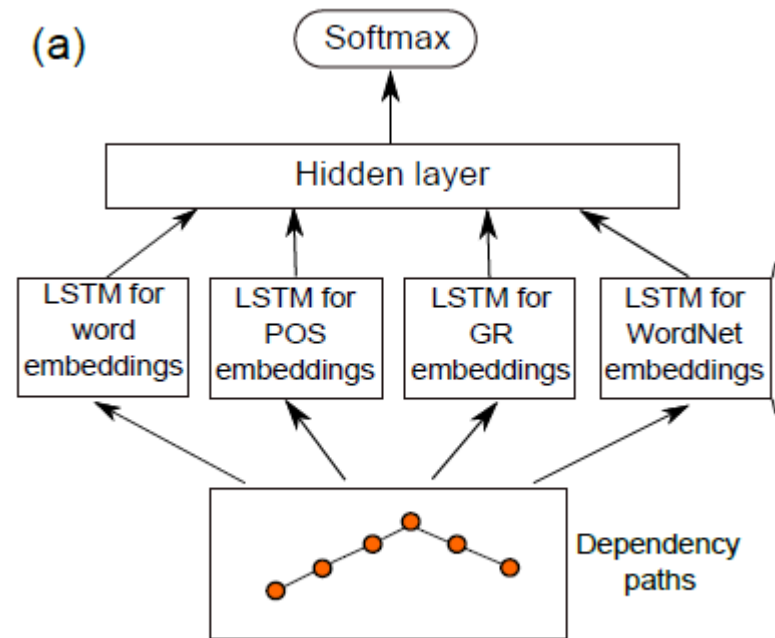
## 02. Model



- Four channels

- SDP를 따라 words, POS tags, grammatical relations, WordNet hypernyms 4개의 정보를 학습한다.

## 02. Model



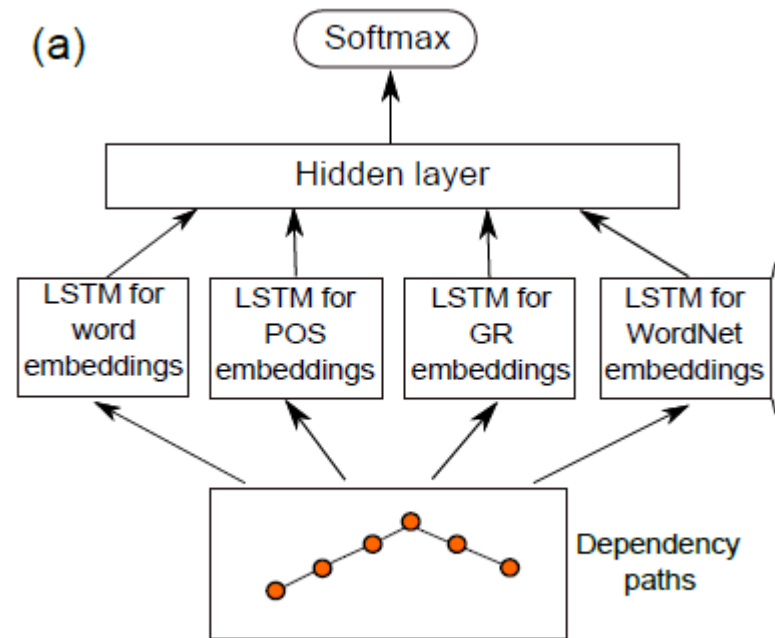
- Four channels(word)

- large corpus에서 단어의 의미를 학습하므로 단어 syntactic and semantic information capture 할 수 있다. (word embedding)

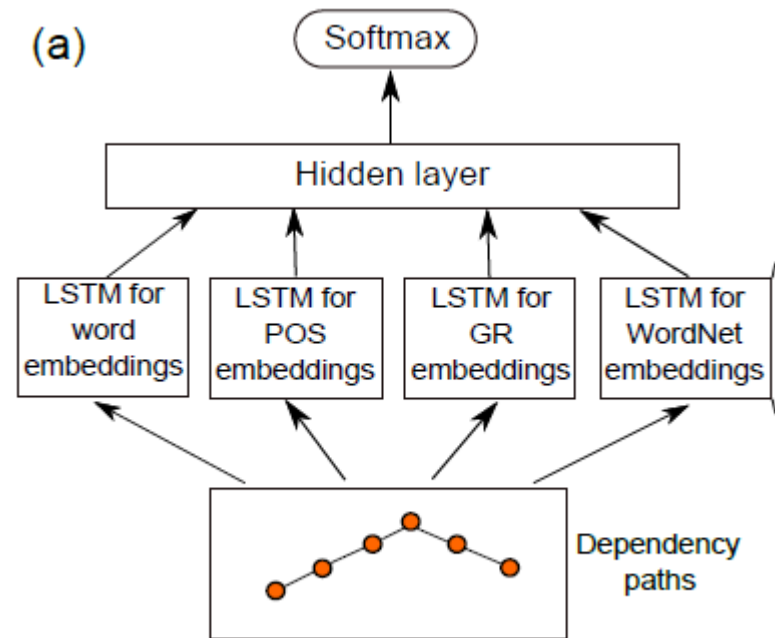
## 02. Model

- Four channels(POS tags)

- word 임베딩은 대규모의 일반 말뭉치에서 학습되기 때문에, word 임베딩이 포함하는 정보는 특정 문장과 일치하지 않을 수 있다.
- 각각의 입력 단어들을 그것의 POS tags, 예를 들어 명사, 동사 등으로 알림으로써 이 문제를 처리했다.



## 02. Model



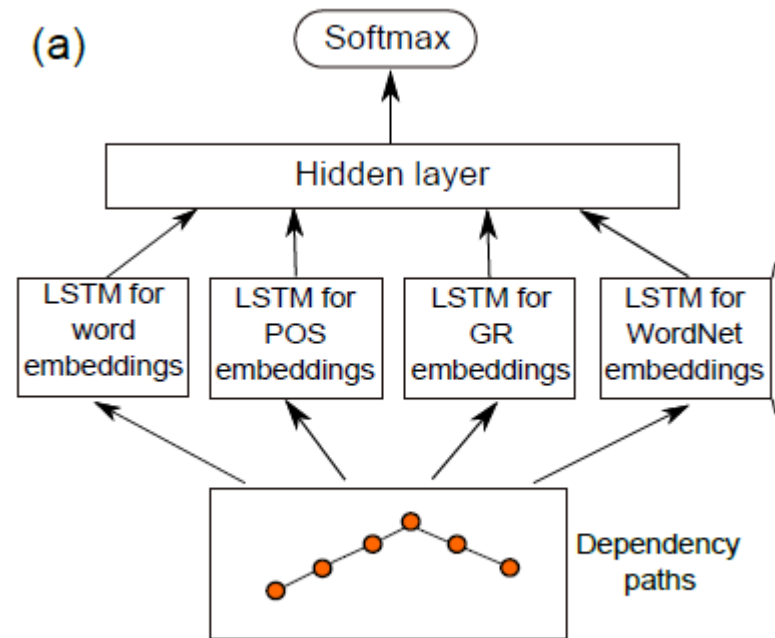
- Four channels(grammaral relations)

- 문장안에서의 단어들의 의존관계는 의미에서 차이를 만든다.

Ex) beats -> it (가주어)

beats -> it (직접 목적어)

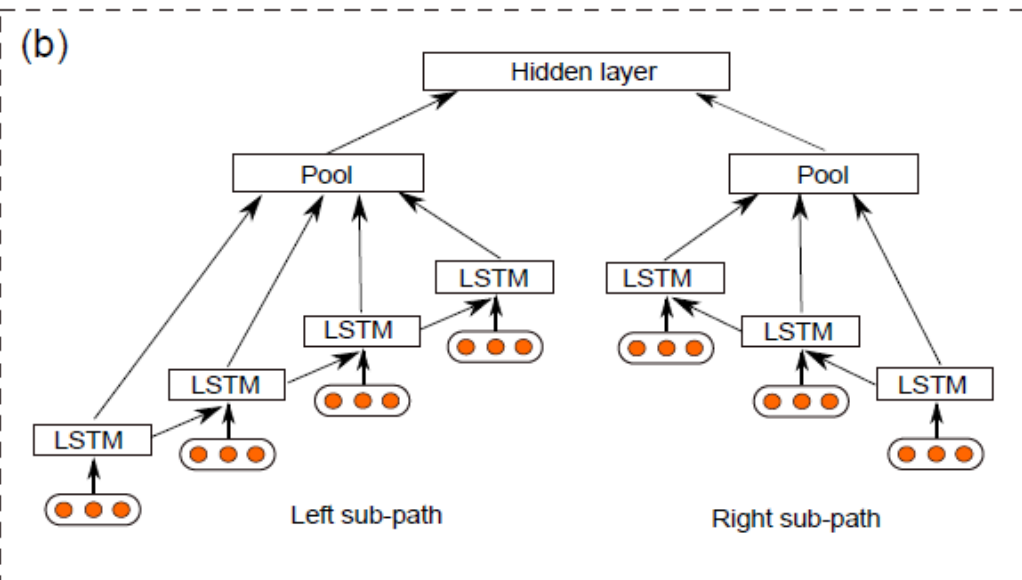
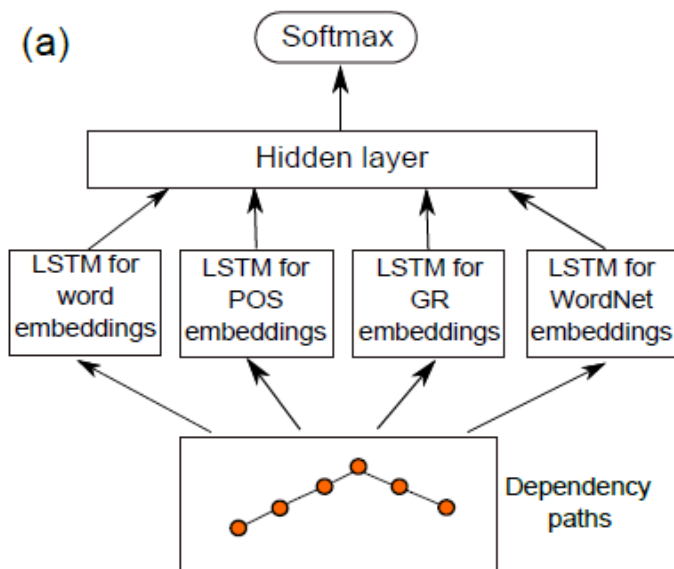
## 02. Model



- Four channels(WordNet hypernyms)

- WordNet의 hypernyms를 사용하면 각각의 단어는 추상적인 개념 정보를 얻을 수 있는데 이것은 서로 다르지만 개념적인 유사 단어들사이의 연결성을 형성하는데 도움을 준다.

## 02. Model



- LSTM

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i)$$

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f)$$

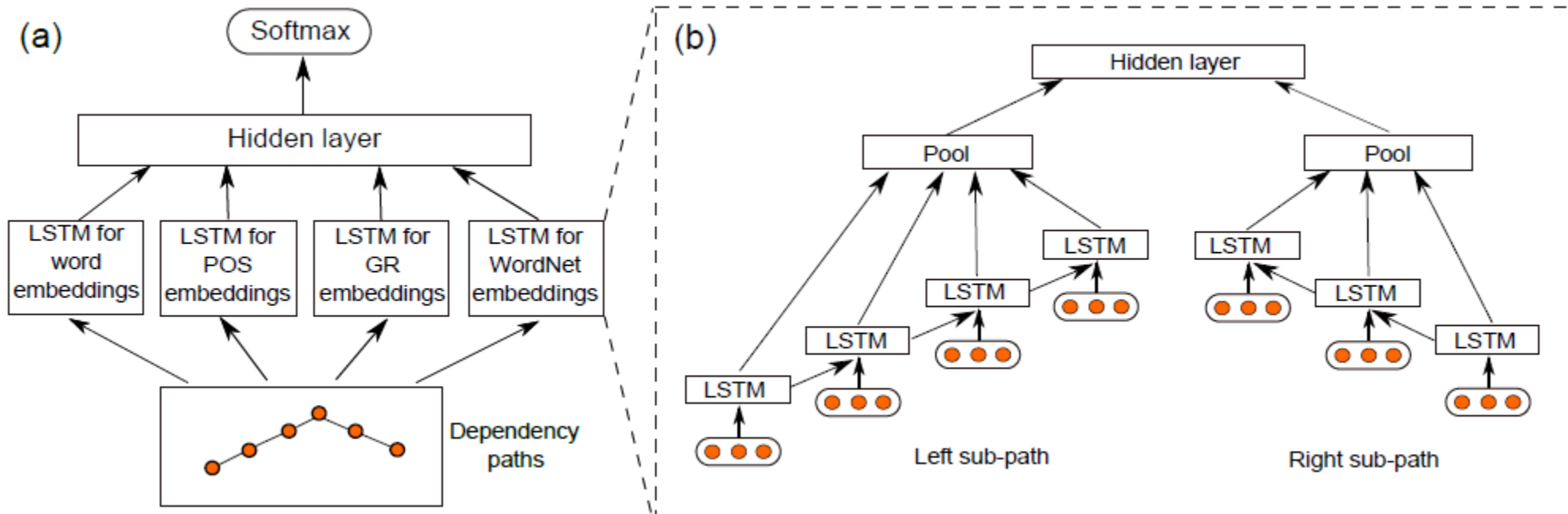
$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o)$$

$$g_t = \tanh(W_g \cdot x_t + U_g \cdot h_{t-1} + b_g)$$

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1}$$

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

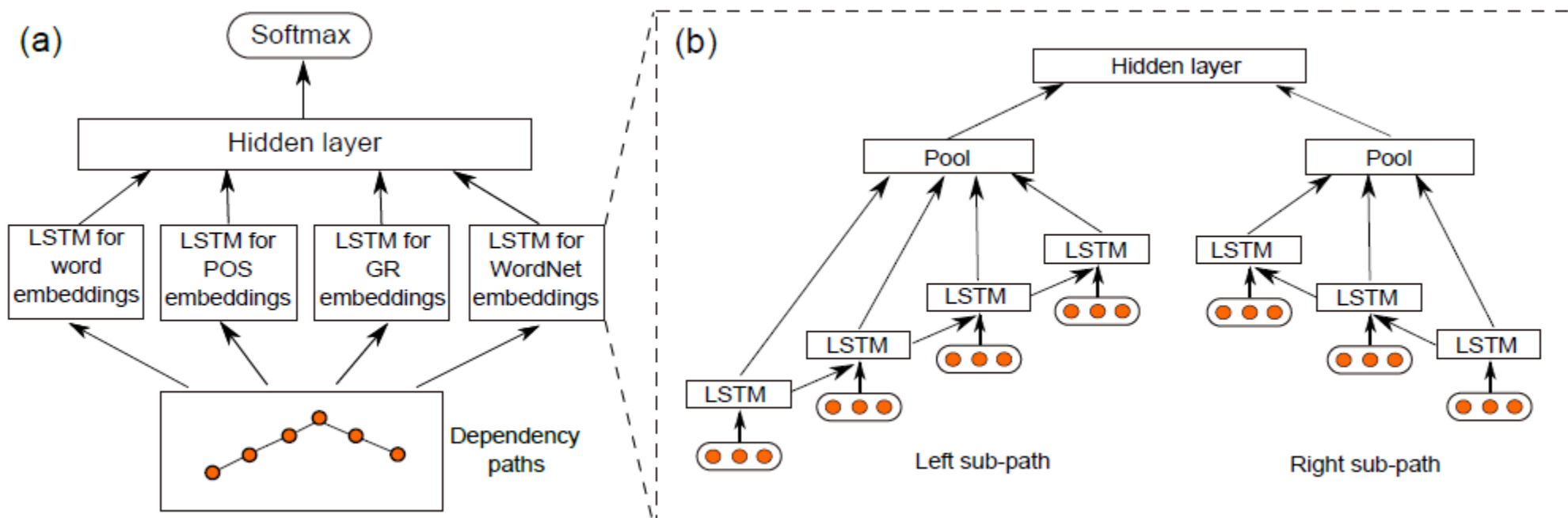
## 02. Model



- **Hidden layer**

- LSTM을 거쳐서 얻은 정보들을 max pooling하고 이렇게 나온 different channels들의 output을 concat 해서 hidden layer에 넣고 softmax를 취한다.

## 02. Model



- Cost function

$$J = - \sum_{i=1}^{n_c} t_i \log y_i + \lambda \left( \sum_{i=1}^{\omega} \|W_i\|_F^2 + \sum_{i=1}^v \|U_i\|_F^2 \right)$$



## 02. Model

---

- Regularization

⇒ 오버피팅을 예방하고 일반화 성능을 높이는 것

- L2 regularization

⇒ 기존의 cost function에 가중치의 제곱을 포함하여 더하는 기법

⇒ 단순히 cost값이 작아지는 방향으로만 진행되는 것이 아니라,  $w$ 값들 역시 최소가 되는 방향으로 진행을 하게 된다.

## 03. Experiments

- Dataset : SemEval-2010 Task 8
- 전체 19개의 라벨이 있는데 방향이 있는 relation 9개 방향이 없는 relation Other class가 존재한다.

## 03. Experiments

Classifier	Feature set	$F_1$
SVM	POS, WordNet, prefixes and other morphological features, dependency parse, Levin classes, PropBank, FanmeNet, NomLex-Plus, Google $n$ -gram, paraphrases, TextRunner	82.2
RNN	Word embeddings	74.8
	Word embeddings, POS, NER, WordNet	77.6
MVRNN	Word embeddings	79.1
	Word embeddings, POS, NER, WordNet	82.4
CNN	Word embeddings	69.7
	Word embeddings, word position embeddings, WordNet	82.7
Chain CNN	Word embeddings, POS, NER, WordNet	82.7
FCM	Word embeddings	80.6
	Word embeddings, dependency parsing, NER	83.0
CR-CNN	Word embeddings	82.8 <sup>†</sup>
	Word embeddings, position embeddings	82.7
	Word embeddings, position embeddings	<b>84.1<sup>†</sup></b>
SDP-LSTM	Word embeddings	82.4
	Word embeddings, POS embeddings, WordNet embeddings, grammar relation embeddings	<b>83.7</b>

Table 1: Comparison of relation classification systems. The “<sup>†</sup>” remark refers to special treatment for the Other class.

## 03. Experiments

Channels	$F_1$
Word embeddings	82.35
+ POS embeddings (only)	82.98
+ GR embeddings (only)	83.21
+ WordNet embeddings (only)	83.03
+ POS + GR + WordNet embeddings	83.70

Table 2: Effect of different channels.

**감사합니다.**