CS224N 2023 | WEEK 1-2 배지섭

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

Neural Networks in NLP (Window classification)

Named Entity Recognition (NER);

문장에서 **개체명(고유명사)을 분류하는 방법론**으로, 문맥 context 를 보고 개체명을 결정한다.

*'…**Museums(개체명)** in Paris are amazing…'*

Foreign	ORG	B-ORG
Ministry	ORG	I-ORG
spokesman	O	O
Shen	PER	B-PER
Guofang	PER	I-PER
told	O	O
Reuters	ORG	B-ORG
that	O	O
:	:	👆 BIO encoding

ORG (조직), O (고유명사가 아님), PER (사람)

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

Neural Networks in NLP (Window classification)

Named Entity Recognition (NER);

문장에서 **개체명(고유명사)을 분류하는 방법론**으로, 문맥 context 를 보고 개체명을 결정한다.

*'…**Museums(개체명)** in Paris are amazing…'*

문장내에서 사용되는 entity 의 정확한 개체를 분류하기 어려운 한계점이 존재

- 동형이의성 (Ambiguity); 동일한 단어도 문맥에 따라 다른 의미를 지님.
- 다양한 표현 방식; 동일한 개체가 다른 단어로 표현되기도 함.

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

Neural Networks in NLP (Window classification)

Named Entity Recognition (NER);

문장에서 **개체명(고유명사)을 분류하는 방법론**으로, 문맥 context 를 보고 개체명을 결정한다.

*'…**Museums(개체명)** in Paris are amazing…'*

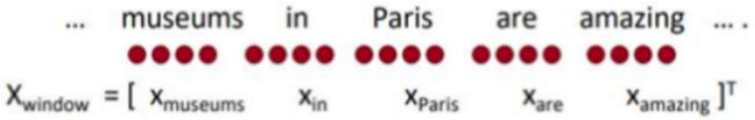
window classification 방법론; 문맥까지 고려

Idea : 중심 단어와 주변 단어들을 함께 분류 문제에 활용하는 방법

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

window classification 방법론; 문맥까지 고려

Idea : 중심 단어와 주변 단어들을 함께 분류 문제에 활용하는 방법



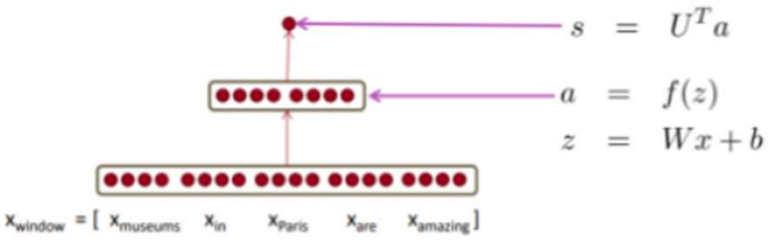
Method1. 단어벡터의 평균으로 계산(위치 정보를 잃어버리는 단점이 존재)

Method2. 벡터를 concatenate 하여 Xwindow 벡터를 만들고 이를 input 으로 취하고
다층퍼셉트론과 softmax clssifier 를 훈련하여 분류를 진행

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

NER Location 분류 예제 : X(paris) 를 Location 으로 분류하는 문제

$$s = U^T f(Wx + b)$$
$$x \in \mathbb{R}^{20\times 1}, W \in \mathbb{R}^{8\times 20}, U \in \mathbb{R}^{8\times 1}$$



score (s) 값에 softmax 를 취해주고 error 함수를 계산하여 W를 업데이트 시킨다.

Max-margin loss(손실함수)

- s = score(museums in Paris are amazing)
- s_c = score(Not all museums in Paris)
- Minimize

$$J = \max(0, 1 - s + s_c)$$

정답과 오답 사이의 거리를 최대로 만드는 margin 을 찾는 방식

$\text{minimize } J = \max(s_c - s, 0)$ Error가 Sc > S 가 아니면 0이 되길 원함.

$\text{minimize } J = \max(\Delta + s_c - s, 0)$ True label – false label > 양수 마진 Δ

즉, (s – sc < Δ)뿐만 아니라 (s – sc < 0)인 경우 오류를 계산하기를 원함.

$\text{minimize } J = \max(1 + s_c - s, 0)$ 마진을 Δ = 1이 되도록 조정

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

Matrix calculus Jacobian matrix : Generalization of Gradient

- It's Jacobian is an ***m* x *n* matrix** of partial derivatives

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)_{ij} = \frac{\partial f_i}{\partial x_j}$$

- Chain Rule

$$z = 3y$$
$$y = x^2$$
$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} = (3)(2x) = 6x$$

$$\boldsymbol{h} = f(\boldsymbol{z})$$
$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$
$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \ldots$$

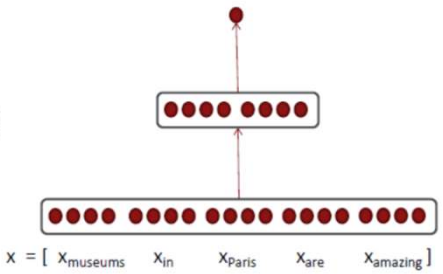
LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

✓ window classification 에서의 역전파 가중치 업데이트 계산과정

$$s = \boldsymbol{u}^T \boldsymbol{h}$$

$$\boldsymbol{h} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

$$\boldsymbol{x} \quad \text{(input)}$$



ds/dW 와 ds/db 를 구해야함!

$$s = \boldsymbol{u}^T \boldsymbol{h}$$
$$\boldsymbol{h} = f(\boldsymbol{z})$$
$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$
$$\boldsymbol{x} \quad \text{(input)}$$

$$\frac{\partial s}{\partial \boldsymbol{b}} = \frac{\partial s}{\partial \boldsymbol{h}} \quad \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \quad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}$$

chain rule 을 적용

편향 b가 벡터의 각 요소에 독립적으로 더해짐.
그 미분이 모든 요소에 대해 1로 동일하게 적용되며,
이는 단위 행렬로 표현

$$\frac{\partial s}{\partial \boldsymbol{b}} = \frac{\partial s}{\partial \boldsymbol{h}} \quad \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \quad \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}$$

$$= \boldsymbol{u}^T \operatorname{diag}(f'(\boldsymbol{z}))\boldsymbol{I}$$

$$= \boldsymbol{u}^T \circ f'(\boldsymbol{z})$$

자코비안 연산 방식을 도입해서 계산한 결과

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

- Using the chain rule again:

$$\frac{\partial s}{\partial \boldsymbol{W}} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$

chain rule 을 다시 적용해서 구하면 됨

$$\frac{\partial s}{\partial \boldsymbol{W}} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$
$$\frac{\partial s}{\partial \boldsymbol{b}} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}$$

The same! Let's avoid duplicated computation...

중복된 계산 (연산량) 을 피하기 위해
파란색 부분 연산을 local error signal 로 묶어서 정의

$$\frac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$
$$\frac{\partial s}{\partial \boldsymbol{b}} = \boldsymbol{\delta} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \boldsymbol{\delta}$$
$$\boldsymbol{\delta} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} = \boldsymbol{u}^T \circ f'(\boldsymbol{z})$$

δ is local error signal

LECTURE 3:
MATRIX CALCULUS AND THE BACKPROPAGATION ALGORITHM

- Using the chain rule again:

$$\frac{\partial s}{\partial \boldsymbol{W}} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$

chain rule 을 다시 적용해서 구하면 됨

$$\frac{\partial s}{\partial \boldsymbol{W}} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$
$$\frac{\partial s}{\partial \boldsymbol{b}} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}}$$

The same! Let's avoid duplicated computation...

중복된 계산 (연산량) 을 피하기 위해
파란색 부분 연산을 local error signal 로 묶어서 정의

$$\frac{\partial s}{\partial \boldsymbol{W}} = \boldsymbol{\delta} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{W}}$$
$$\frac{\partial s}{\partial \boldsymbol{b}} = \boldsymbol{\delta} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{b}} = \boldsymbol{\delta}$$
$$\boldsymbol{\delta} = \frac{\partial s}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}} = \boldsymbol{u}^T \circ f'(\boldsymbol{z})$$

δ is local error signal

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

문법 태깅

태그	설명	예시
nsubj	명사 주어	She (**nsubj**) loves ice cream.
obj	목적어	She loves (**obj**) ice cream.
iobj	간접 목적어	He gave (**iobj**) her a book.
ccomp	절적 보어	She said (**ccomp**) that she would come.
xcomp	개방형 절적 보어	She wants (**xcomp**) to leave.
advmod	부사적 수식어	She runs (**advmod**) quickly.
amod	형용사적 수식어	The (**amod**) red car.
nmod	명사적 수식어	The book (**nmod**) on the table.
det	한정사	(**det**) The book.
conj	접속된 어구	She likes tea (**conj**) and coffee.
cc	등위접속사	She likes tea and (**cc**) coffee.

POS; part of speech, 품사

POS 태그	설명	예시
NN	명사	cat, book, happiness
VB	동사	run, eat, be
JJ	형용사	happy, red, large
RB	부사	quickly, very, well
PRP	대명사	he, she, it, they
DT	한정사	the, a, an, this
IN	전치사	in, on, at, by
CC	등위접속사	and, but, or
MD	조동사	can, will, should
RP	불변화사	up, off, out (give up, take off)

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

Parsing ; 문장의 문법적 구조를 파악하는 과정

구문 분석 (Consititency parsing);
문장의 구조를 파악하는 방법으로 직관적으로 문장의 구성요소들을 파악을 하는 방법
**주로 영어권 단어에 사용되는 문장 분석 방법

의존성 분석 (Dependency parsing) ;
단어 간의 의존 관계를 따져서 구조를 파악하는 방법

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

Parsing ; 문장의 문법적 구조를 파악하는 과정

구문 분석 (Consititency parsing);
문장의 구조를 파악하는 방법으로 직관적으로 문장의 구성요소들을 파악을 하는 방법
**주로 영어권 단어에 사용되는 문장 분석 방법

Starting unit: words

the, cat, cuddly, by, door

Words combine into phrases

the cuddly cat, by the door

Phrases can combine into bigger phrases

the cuddly cat by the door

*"the cat cuddly **by door**"*
명사구(Noun Phrase) 전치사구(Preprositional Phrase)

문장 분석을 위해
"문법 규칙을 사용하여 구문을 조합하고 결합하여 문장을 만들어내는 것"
Context-Free Grammars(CFGs)를 사용한 방법

문법 규칙에 의해 '명사구'와 '전치사구'를 정의하고 결합하여 더 큰 의미를 가진 구문을 재귀적으로 반복해 언젠가 문장을 형성

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

의존성 분석; Dependency parsing, 단어 간에 가지고 있는 문법적 변화를 파악한다.

'The cats'가 어떤 행동을 하는지 결정

"The cats scratch people with claws"

'The cats'는 'scratch'라는 행동의 주체

'people'에게 행하는 행위

"The cats scratch people with claws"

'scratch'가 중심어('head', 'governer')이고
명사 'cats'와 'people'이 수식어('dependent', 'modifier')

- 중심어(governor): 문장에서 중요한 단어, 보통
- 수식어(dependent): 중심어를 보충하거나 설명하는 단어.

의존성 분석을 통해 해당 문장의 관계를 다음과 같이 표현할 수 있음.

- "Scratch " (동사) ← ROOT (문장)
- "cats" (주어) ← "scratch" (동사)
- "scratch" (동사) → "people" (목적어)
- "scratch" (동사) → "with claws" (수식구)

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

의존성 분석; Dependency parsing, 단어 간에 가지고 있는 문법적 변화를 파악한다.

중심어를 어떻게 선택을 해야하는 지를 파악이 안되는 경우가 더 많을 수 있다.

그렇다면 다음의 기준을 따라보자.

1. 중심어(head)는 구성의 통사적 범주를 결정한다. 예를 들어, 명사는 명사구의 중심어이고, 동사는 동사구의 중심어이다.
2. 수식어(modifier)는 선택적일 수 있으나, 중심어는 필수적이다
   예를 들어, 문장 "cats scratch people with claws"에서
   하위 트리 "cats scratch"와 "cats scratch people"은 문법적으로 올바른 문장이지만, "with claws"는 그렇지 않다.
3. 중심어는 수식어의 형태적 특성을 결정한다. 예를 들어, 성별 일치를 요구하는 언어에서는 명사의 성별이 형용사와 한정사의 성별을 결정한다.
4. 우선 실질어(content words)를 연결한 후, 기능어(function words)를 연결한다.

실질어(실질적 의미를 전달하는 단어): 실질어는 주로 명사, 동사, 형용사, 부사 등을 포함한다.
(예: 명사(cat, house), 동사(run, eat), 형용사(happy, blue), 부사(quickly, silently))
기능어(문법적 역할을 하는 단어): 기능어는 문법적 기능을 수행하며, 문장의 구조 를 형성하고 연결하는 데 도움을 준다.
(예: 관사(the, a), 대명사(he, she), 전치사(in, on), 접속사(and, but), 조동사(can, will))

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

의존성 분석; Dependency parsing, 단어 간에 가지고 있는 문법적 변화를 파악한다.

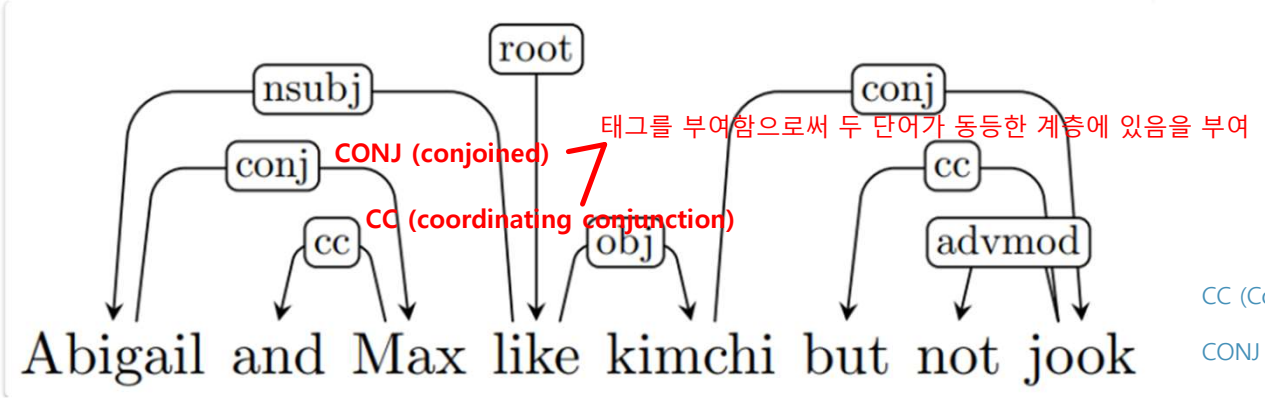
"Abigail and Max like gimchi"

Abigail와 Max, *Abigail and Max* 중 어떤 단어를 수식할까?

Abigail and Max'를 수식해야 하지만 실질어 사이에 기능어 and가 존재하기 때문에 위의 가이드라인에 위반

수식어를 누구에게 선부여를 해야 하는가?

둘은 우위가 존재하지 않기에 둘 다 부여를 해야 한다.

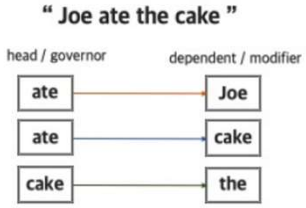


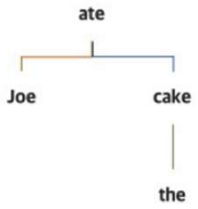
태그를 부여함으로써 두 단어가 동등한 계층에 있음을 부여

CC (Coordinating Conjunction): 등위접속사 자체를 나타내는 태그
주로 접 속사인 "and", "but", "or" 등을 가리킴
CONJ (Conjoined): 접속사에 의해 연결된 구나 절을 나타내는 태그.

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

의존성 분석; Dependency parsing, 단어 간에 가지고 있는 문법적 변화를 파악한다.





Dependency parsing의 보편적 특성

- Bilexical affinities: 두 단어 사이의 실제 의미가 드러나는 관계를 의미
- Dependency distance: 의존 관계에서 두 단어 사이의 거리를 의미
- Intervening material: 마침표, 세미콜론과 같은 구두점을 넘어서는 의존 관계는 형성되지 않음을 의미
- Valency of heads: 중심어(head)가 좌우측에 몇 개의 의존어(dependents)를 가질 것인가에 대한 특성

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

컴퓨터가 문장을 분석하기 위한 알고리즘

- 전이기반 의존성 분석 (Transition-based dependency parsing)
- 그래프 알고리즘 (graph algorithm)

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

컴퓨터가 문장을 분석하기 위한 알고리즘

- 전이기반 의존성 분석 (Transition-based dependency parsing)

각 구성(configuration)은 세 요소 (σ, β, A) 로 표현되며
Σ 는 스택(stack)
β 는 입력 버퍼(input buffer)
A 는 생성된 arc들의 집합
을 의미한다.

전체 입력은 버퍼에 있다.

$C_{initial} = ([ROOT], w, \emptyset)$

Σ 스택에는 특별한 노드 ROOT 만 있다. arc 집합은 비어 있다.

버퍼는 비어 있다

$= ([ROOT], \emptyset, A)$

스택에는 ROOT 만 남아있다. arc들이 완전한 트리를 형성한 상태이다.

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

컴퓨터가 문장을 분석하기 위한 알고리즘

- 전이기반 의존성 분석 (Transition-based dependency parsing)

σ	β	action	arc added to A
[ROOT]	She loves ice cream	SHIFT	–
[ROOT, She]	loves ice cream	SHIFT	–
[ROOT, She, loves]	ice cream	ARC-LEFT	She $\leftarrow$ loves
[ROOT, loves]	ice cream	SHIFT	–
[ROOT, loves, ice]	cream	SHIFT	–
[ROOT, loves, ice, cream]	$\emptyset$	ARC-RIGHT	ice cream $\rightarrow$ cream
[ROOT, loves, ice]	$\emptyset$	ARC-RIGHT	loves $\rightarrow$ ice cream
[ROOT, loves]	$\emptyset$	ARC-RIGHT	ROOT $\rightarrow$ loves
[ROOT]	$\emptyset$	DONE	–

전이 규칙: 입력 버퍼의 첫 번째 단어를 스택으로 이동.

전이 규칙: 입력 버퍼의 첫 번째 단어를 스택으로 이동.

전이 규칙: 스택의 두 번째 항목("She")을 스택 맨 위 항목("loves")의 의존어로 설정, 스택에서 두 번째 항목(의존어) 제거.

전이 규칙: 입력 버퍼의 첫 번째 단어를 스택으로 이동.

전이 규칙: 스택의 맨 위 항목("cream")을 스택의 두 번째 항목("loves")의 의존 어로 설정하고, 스택에서 맨 위 항목을 제거.

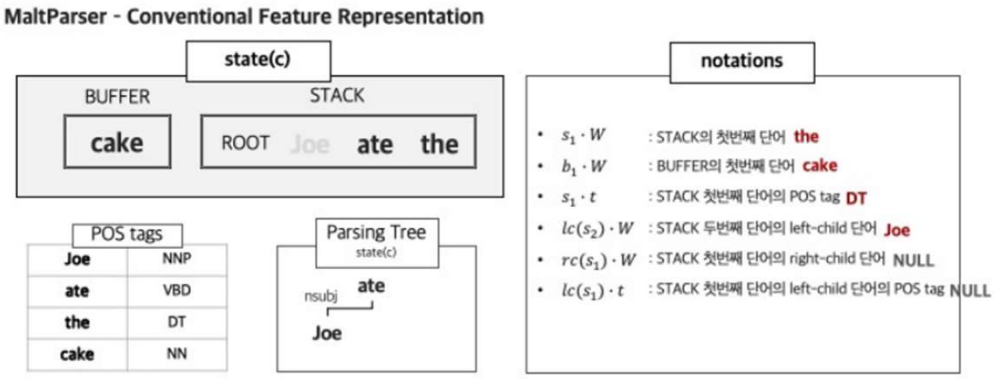
현 시점의 입력 버퍼와 스택 그리고 아크 집합으로 이뤄진 상태(state) 기반으로 특정한 전이 규칙을 따르는 함수를 통해서 전이 규칙을 정하는 것

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

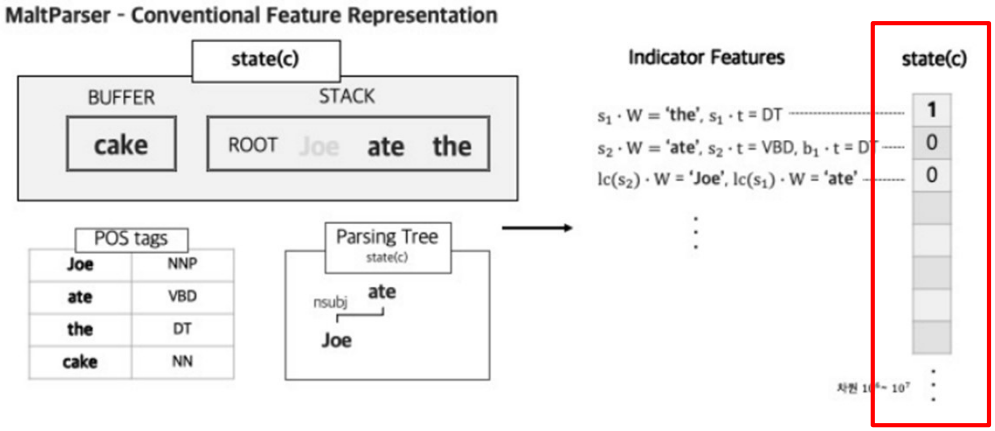
컴퓨터가 문장을 분석하기 위한 알고리즘

- 전이기반 의존성 분석 (Transition-based dependency parsing)

상태를 어떻게 임베딩 하는 것이 나을까?



각 토큰의 테그를 활용



특정 조건을 만족하면 '1' 아니면 '0'으로 나타낼 수 있는 Indicator Feature를 이용

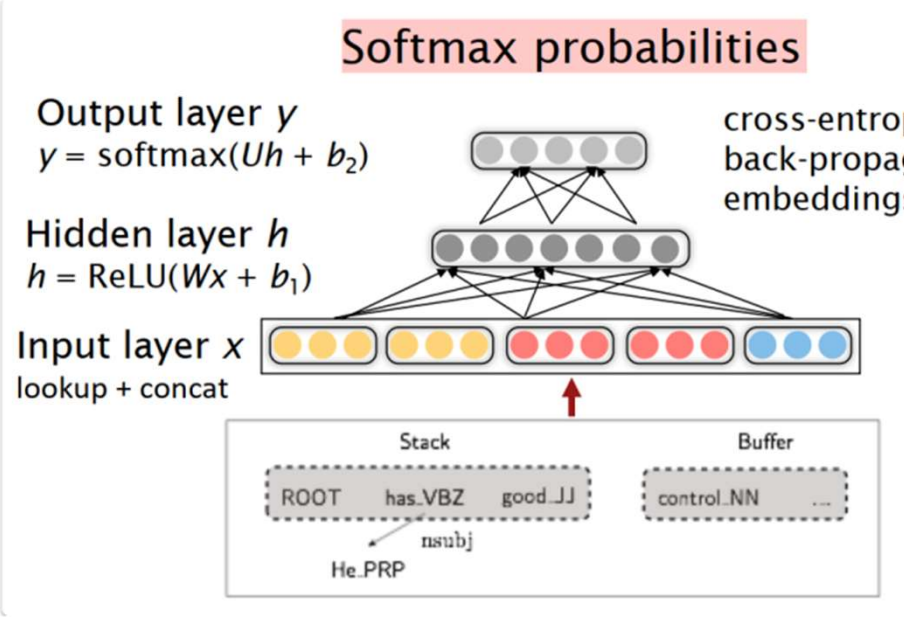
이러한 모든 조건들을 만족하는 벡터가 굉장히 희소한 형태라는 문제점이 발생

단어 본래의 의미 또는 태그의 의미에 대한 학습이 제대로 이뤄지지 못 한다

LECTURE 4:
LINGUISTIC STRUCTURE DEPENDENCY PARSING

컴퓨터가 문장을 분석하기 위한 알고리즘

- **Neural Denpendency parsing** 전이 기반 의존성 분석을 신경망을 적용하여 의존성 분석을 통해 문장의 구조 분석 성능을 끌어올릴 수 있도록 하는 기법



words feature로 들어가게 되는 데이터는 총 18개로 구성

STACK과 BUFFER의 TOP 3 words (6개)
STACK TOP 1, 2 words의 첫번째, 두번째 left & right child word (8개)
STACK TOP 1,2 words의 left of left & right of right child word (8개)

arc labels에서는 STACK과 BUFFER의 TOP 3 words 6개를 제외한 12개의 label(dependent 관계 표시) 데이터로 구성

입력으로 들어가는 feature는 words, POS tag(태그) , arc labels 3가지로 구분