전산언어학 소개:

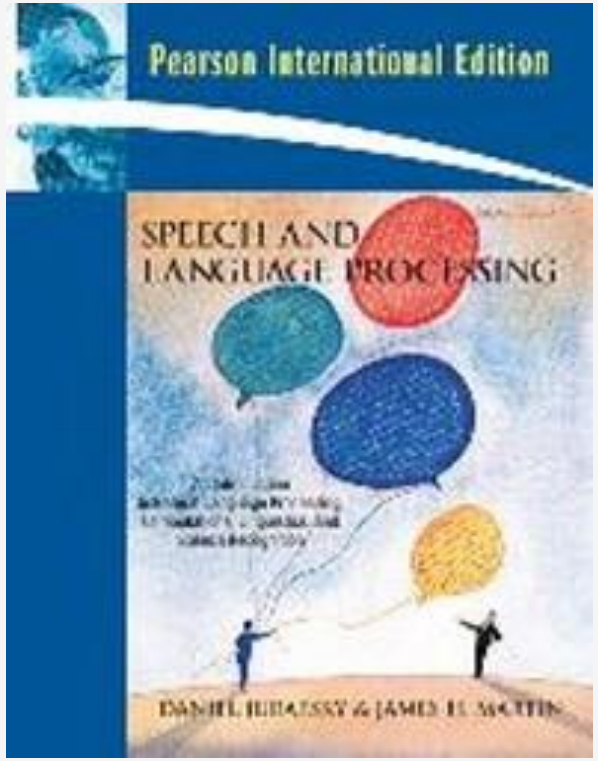
기본 개념, 역사, 현재, 그리고 미래

2020년 5월 4일

김영삼 (youngsamy@gmail.com)

강사 & 책 소개

- 이름: 김영삼
- 최종학력: 서울대학교 인지과학과정 공학박사 (전산언어학 전공)
- 관심 주제: 강화학습, 대화시스템, 지식표현 및 추론 시스템
- 근무경력:
  텍스트팩토리 (2017)
  서울대 시간강사 (2018)
  CELI Inc. (2019)
  삼성생명 디지털 추친팀 (챗봇개발 그룹)
- NLP 교과서
  - Speech and Language Processing (Jurafsky & Martin)



Content

- 전산언어학 혹은 자연어 처리(NLP) 주요 개념 소개
- 역사적 발전 과정에 대한 개략적 소개
- 현재 활발히 연구되는 주요 방법론 소개
- 미래에 대한 짧은 조망

How to question?

- 한 슬라이드의 내용이 거의 끝나거나 끝난 다음에 질문해주세요.
- 쉬는 시간 직전과 맨 수업 말미에 Q&A 시간이 있습니다.
- 이전 슬라이드 내용에 대한 질문시 우측 하단의 슬라이드 번호를 메모해주세요.

전산 언어학 = 전산적 언어학

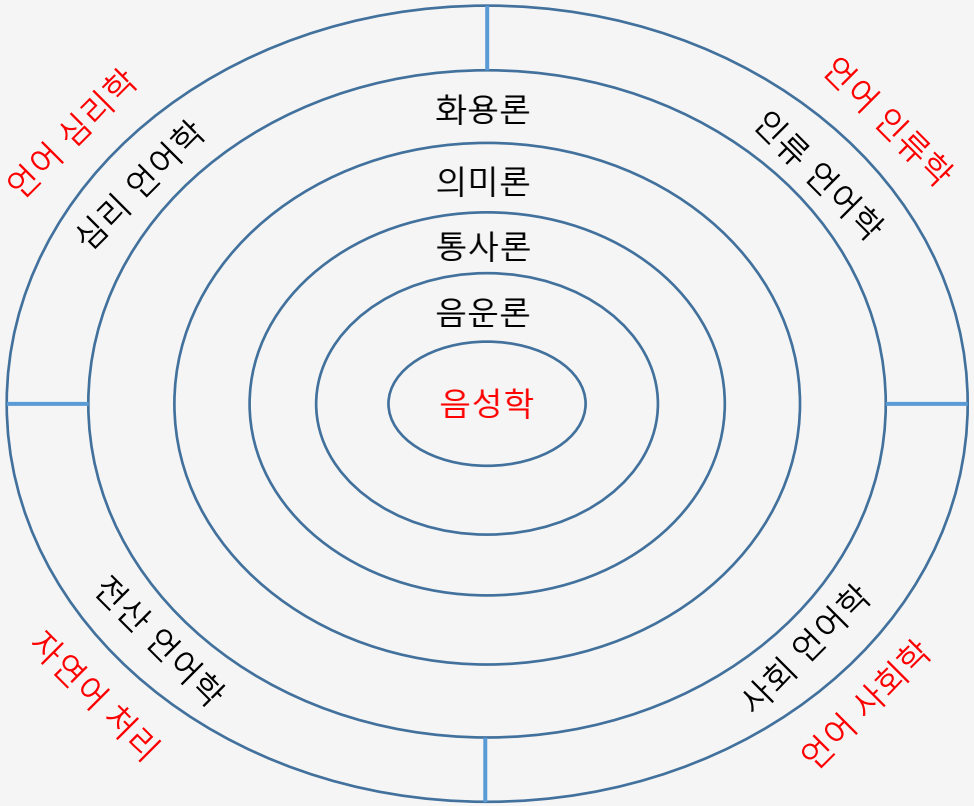
전산적 (computational)

계산적 방법을 이용한다는 뜻. 계산이란 **알고리즘**을 진행하는 과정을 의미한다.
알고리즘이란 명시적으로 표현된 유한한 길이의 절차를 말한다.
컴퓨터는 알고리즘으로 표현 가능한 모든 계산을 실행할 수 있다.
즉, 전산적이란 말은 알고리즘을 통해 주어진 문제를 풀려는 접근법을 뜻함.

언어학 (linguistics)

언어의 본질구조에 대해 **연구**하는 학문.
여기서 연구는 관찰, 가설제시, 예측과 같은 과학적 방법을 가리킨다.
언어학은 광범위한 분야를 갖고 있다.
가장 큰 단위로 언급하면, **음운론**, **통사론**, **의미론**, **화용론**으로 이루어져 있다.

언어학과 그 인접학문들


화용론
의미론
통사론
음운론
음성학
심리 언어학
인류 언어학
전산 언어학
사회 언어학
언어 심리학
언어 인류학
자연어 처리
언어 사회학

음운론이란 무엇인가?

- 언어의 소리 체계를 연구한다.
- 음성학과 다른 점은 말소리의 물리적인 요소가 아니라 언어적 소리의 특성을 다룬다.
- 따라서 언어 내 소리의 변별자질, 즉 음소(phoneme)에 대한 연구가 주를 이룬다.
- 음소는 크게 두 가지 유형, 자음과 모음으로 구분되며 모음은 불변모음과 이중모음으로 흔히 구분된다.
- 음운론적 질문의 예:

  한국어의 모음 '에'와 '애'는 서로 구별가능한 음소일까?

- 그 외에 변이음, 소리결합, 비분절 음소, 운율적 음운 등이 주요 연구대상이다.

형태론과 통사론

- 형태론(morphology)은 단어의 어형변화에 대한 문법을 연구한다.
- 단어를 구성하는 가장 작은 통사적 단위를 형태소(morpheme)라고 한다.

The	sheep	walk	ing	albatross	chant	ed	a	dream	y	lullaby
1	2	3	4	5	6	7	8	9	10	11

- 단어의 형태론은 크게 세 가지 유형으로 나뉜다.
  - 고립형: 단어가 더 작은 단위로 나뉘지 않는다. (중국어)
  - 교착형: 단어가 더 작은 형태소 단위로 나뉜다. (한국어, 일본어, 터키어)
  - 굴절형: 형태소 사이의 경계가 분명하지 않다. (라틴어, 독일어)
- 모든 언어에는 유한한 수의 단어 유형이 있고, 이것을 품사(parts of speech)라고 부른다. (명사, 동사, 형용사, 전치사 등이 대표적)

구문론 혹은 통사론

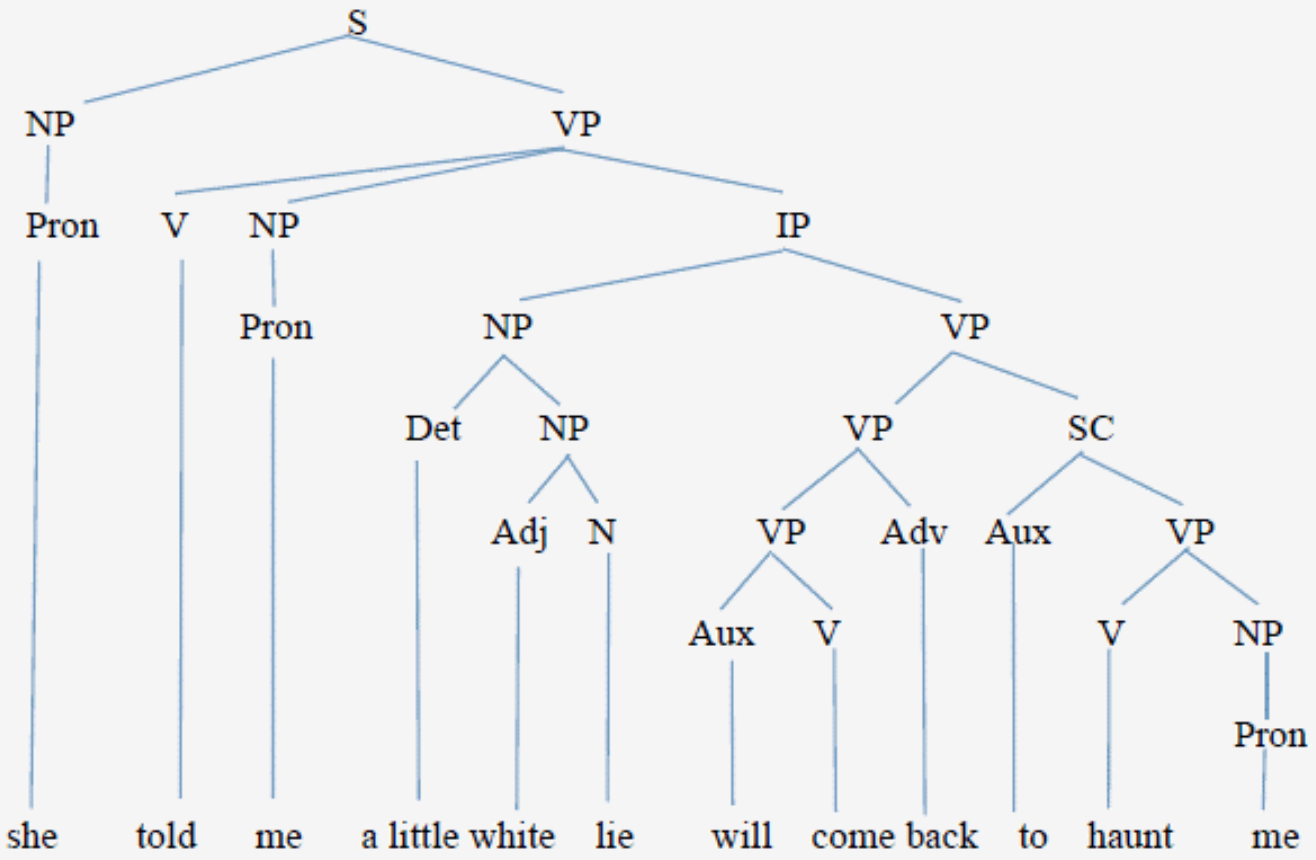
- 형태론이 단어 내 요소들의 문법을 연구한다면 통사론은 문장 내 요소들의 문법을 연구한다.

- 모든 언어는 단어들을 서로 연결시키는 다양한 방식을 사용한다.
  - 어순 (word order)
  - 굴절 (-ing, -ed, -s, -es)
  - 기능어 (of, by, that)
  - 조사 (은/는/이/가/을/를)

- 문장은 주부와 술부처럼 어떤 구성성분(constituents)으로 나누어 분석할 수 있다.

- 이 구성성분을 나무 구조로 표상한 것은 수형도라고 부르며 이는 통사론의 주요 분석방법이다.

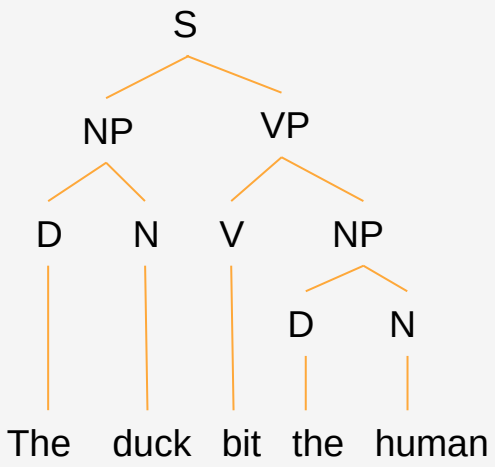
문장의 수형도 예제


S
NP
VP
Pron
V
NP
IP
Pron
NP
VP
Det
NP
VP
SC
Adj
N
VP
Adv
Aux
VP
Aux
V
V
NP
Pron
she
told
me
a little white
lie
will
come back
to
haunt
me

다시쓰기 규칙

- 수형도를 표현하는 형식적인 방법은 다시쓰기(rewrite) 규칙을 이용하는 것이다.

S -> NP VP
VP -> V NP
NP -> D N
N -> duck, human
V -> bit
D -> the



통사론, 의미론, 화용론

- 연구 목표들
  - 통사론: 단어들 간의 형식적 관계
  - 의미론: 언어적 표현이 가리키는 대상과 표현과의 관계
  - 화용론: 화자와 청자가 연관된 문장과 의미의 관계

- 의미론과 화용론의 경계는 애매하다.

- 통상적 정의에 따르면 의미론은 언어적 의미에만 초점을 맞추고 세계지식에서 비롯되는 의미는 연구 대상으로 삼지 않는다.

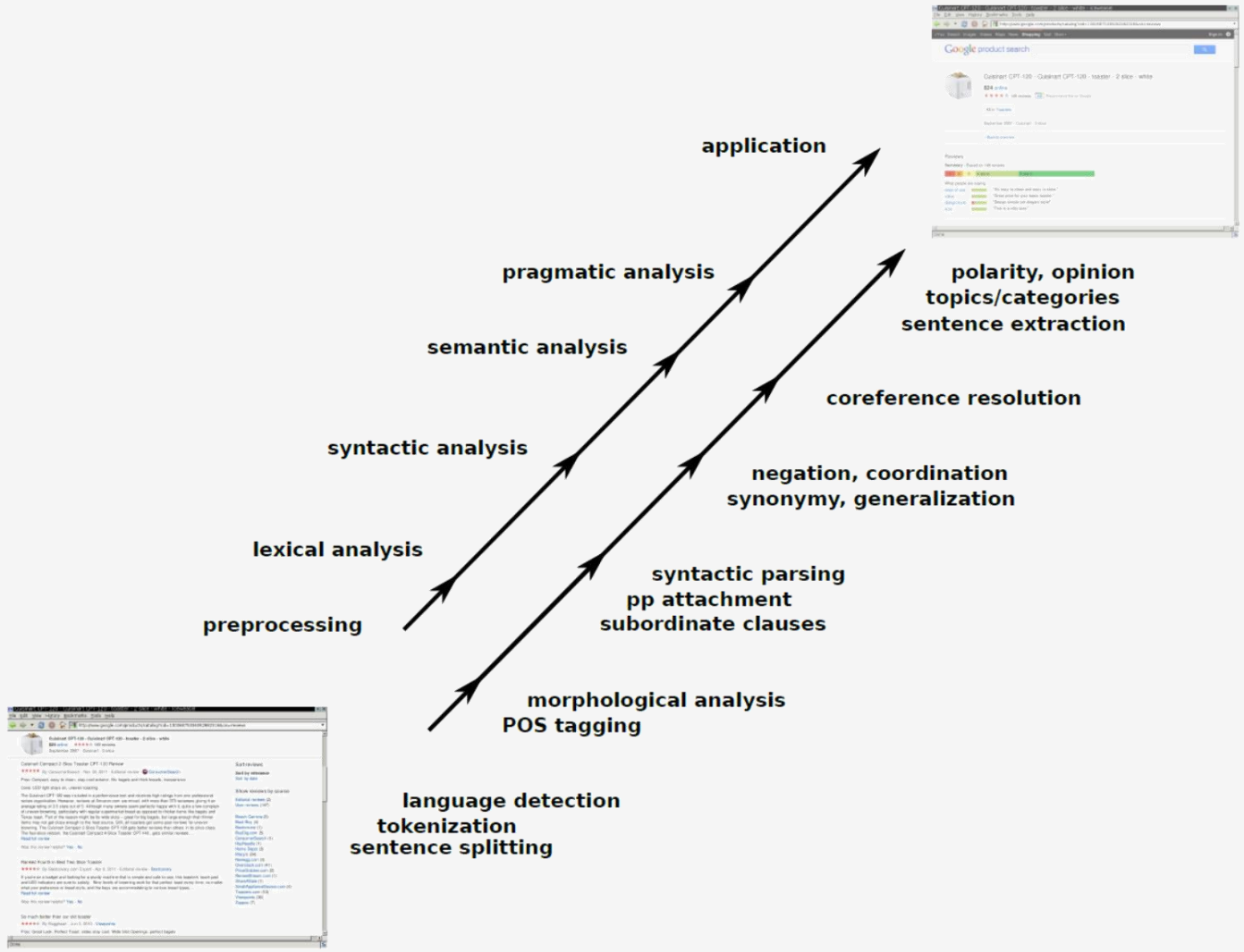
- 따라서 의미론은 형식적 의미, 혹은 문장의 진리조건을 의미의 중요한 초석으로 삼는다.

- 반면에 화용론은 의사소통에 관여하는 의미를 연구하며 함축, 전제, 화행 등이 주요 연구분야이다.

NLP Pipeline

- NLP 시스템의 표준적 아키텍쳐
- 단순 전처리부터 시작해 점점 복잡한 task들로 이어진다.
- 엄밀히 따지면 파이프라인의 각 단계들은 서로 독립적이지 않다.
- 하지만 이 단계들을 모듈로 분리하여 시스템의 유지 및 개선의 이점을 누린다.

NLP analyses and NLP tasks in parallel line



Text Preprocessing

- The task of converting raw text files (HTML, XML, PDF, etc.) into well-defined sequence of linguistically meaningful units:
  - Characters
  - Words
  - Sentences

- Character encoding

- Language identification

- Text segmentation:
  - Segmentation from images, tables, html formats, etc.
  - Sentence segmentation
  - Paragraph segmentation

Character Encoding

- 텍스트 처리의 가장 원초적인 단계
- 8비트 character sets: ASCII, ISO-8859(유럽 언어들)
- 2-byte character sets: 중국어, 일본어 같은 보다 큰 글자들을 가진 언어들을 포함
- Code-switiching: 한 문서 안에서 다른 글자 인코딩 셋이 사용된 경우
- 현재 가장 널리 이용되는 국제 글자 인코딩 셋은 UTF-8 이다.
  - MS Windows는 한국어에 대해 독자적 인코딩 셋인 cp-949 이용
  - 그 외에도 euc-kr 셋도 있으나 최근엔 잘 쓰이지 않는다.
- 한글의 경우 조합형을 사용하지 않고 완성형 인코딩 셋을 쓰므로 초중성 구분을 위해서는 인위적 변환이 필요하다.

Language Identification

- 같은 인코딩 셋을 쓰는 경우 각 언어는 글자 인코딩 번호에 의해 쉽게 구분 가능하다.
- 일반적으로 글자 인코딩 셋은 언어권 단위로 구축된 경우가 많다.
  - 라틴어권, 한자문화권(CJK), 러시아어권, 노르웨이권 등
- 글자 인코딩 셋에는 일반적으로 character set에 대한 메타정보를 기록하지만 그렇지 않을 경우 byte 정보의 분포를 통해 구분해야 한다.

```
>>> import urllib
>>> rawdata = urllib.urlopen('http://yahoo.co.jp/').read()
>>> import chardet
>>> chardet.detect(rawdata)
{'encoding': 'EUC-JP', 'confidence': 0.99}
```

Text Segmentation

- 컴퓨터에서 텍스트는 다양한 어플리케이션을 통해 작성되고 각 어플리케이션은 다양한 포멧을 통해 텍스트를 인코딩하여 저장한다.
- 자연어 텍스트를 이미지나 테이블, 프로그래밍 언어 등과 구분하여 추출하는 과제
- 대표적인 경우가 HTML으로부터 텍스트를 추출하는 과제이다.
- Example using NLTK library

```
>>> url = "http://news.bbc.co.uk/2/hi/health/2284783.stm"
>>> html = urlopen(url).read()
>>> html[:60]
'<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN'

>>> raw = nltk.clean_html(html)
>>> tokens = nltk.word_tokenize(raw)
>>> tokens
['BBC', 'NEWS', '|', 'Health', '|', 'Blondes', "'", 'to', 'die', 'out', ...]
```

Tokenization

- Type과 token 구분
  - 자연어 처리에서 단어의 type은 그 고유한 어휘를 가리킨다.
  - 단어의 token은 나타난 특정 단어를 가리킨다.
  - “단어, 단어, 단어” → type 수: 2, token 수: 5
- 공백을 통해 tokenization이 쉬운 경우와 어려운 경우
  - 쉬운 언어들: 대부분의 유럽 언어들
  - 어려운 언어들: 중국어, 한국어, 터키어
  - 중간정도 어려운 언어들: 라틴어, 독일어

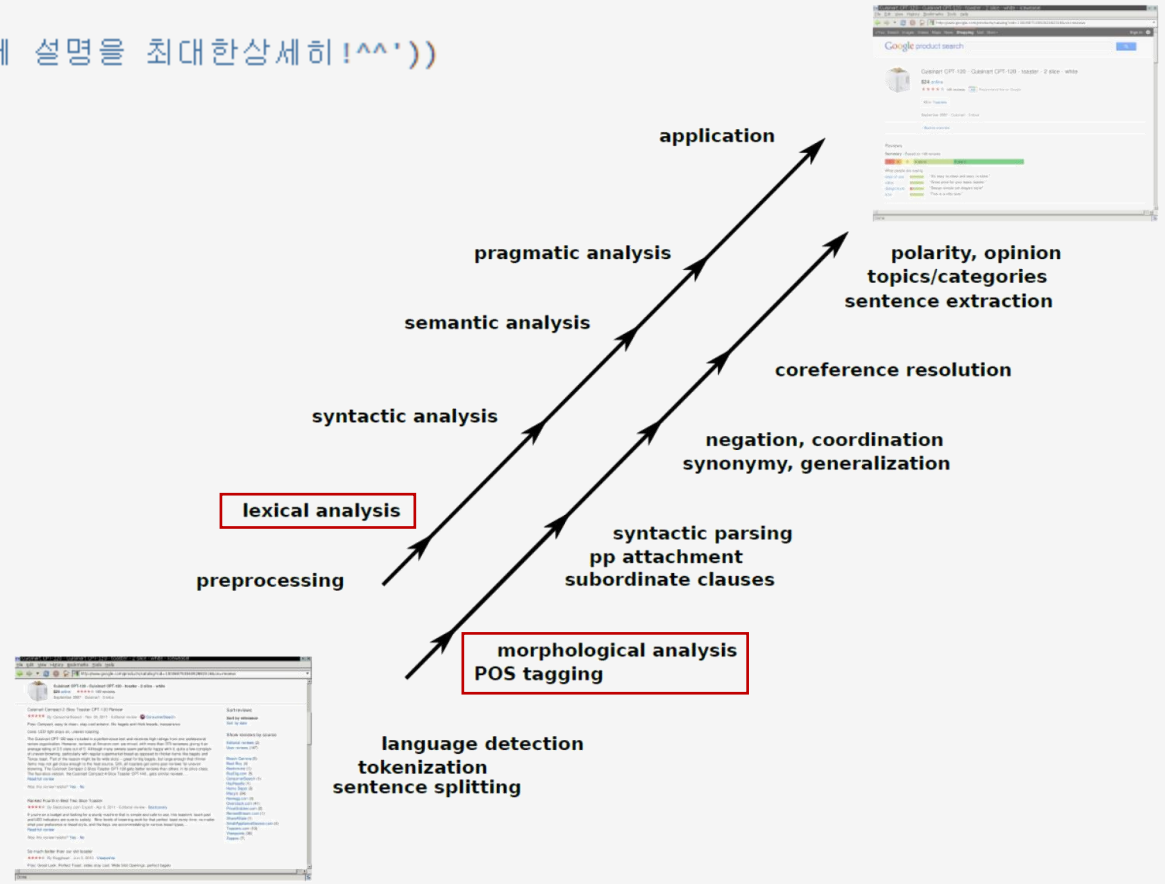
Lexical Analysis

단어는 자연어 텍스트의 가장 기초적인 요소이고 언어학의 형태론은 단어가 형태소로부터 어떻게 구성되는지에 대한 것이다.

- Lemmatization: map the word to its root
- Stemming: retain the stem, throw away suffixes
- Parts-of-Speech (POS): word classes
  - POS-tagging (형태소 분석): 각 단어들에 품사 태그를 부여한다.
  - 한국어 품사 태그는 태깅 체계에 따라 상이하다. (태그셋 비교표)

한국어 형태소 분석 예제

```
>>> from konlpy.tag import Kkma
>>> from konlpy.utils import pprint
>>> kkma = Kkma()
>>> pprint(kkma.pos(u'오류보고는 실행환경, 에러메세지와함께 설명을 최대한상세히!^^'))
[(오류, NNG),
 (보고, NNG),
 (는, JX),
 (실행, NNG),
 (환경, NNG),
 (,, SP),
 (에러, NNG),
 (메세지, NNG),
 (와, JKM),
 (함께, MAG),
 (설명, NNG),
 (을, JKO),
 (최대한, NNG),
 (상세히, MAG),
 (!, SF),
 (^^, EMO)]
```



형태소 분석의 어려움

- 중의성 발생
  - 형태소 부여는 맥락-의존적이다.
    - Case A:
      Q: What are they doing in the kitchen?
      A: They are cooking apples. (POS-tag: verb-ing)
    - Case B:
      Q: What kind of apples are those?
      A: They are cooking apples. (POS-tag: adj)
- 미지어(unknown words) 발생
  - 자연어에서는 언제나 신조어가 발생한다.
- 태그셋의 일관성
  - 품사 태그셋 또한 변할 수 있다.
  - 정밀한 태그셋은 분석에는 용이하지만 태깅작업은 어렵게 만든다.

Syntactic Analysis

POS 태그셋의 연쇄열만으로는 문장이 어떻게 구성되는지를 이해할 수 없다.

이제 우리는 **문법(grammar)**이 필요하다.

- Constituency: 단어집단은 마치 하나의 유닛처럼 움직인다. (e.g., 이화 여자대학교, 대한민국에서 제일 큰 기업)
- Grammatical relations: 문장 내에서 단어와 단어가 서로 맺는 관계 (주어와 술어 관계 등)
- Dependency relations: 통사/의미 정보를 통해 통사적 관계를 제약한다.
  - “The ham sadwitch next to the man studies computer science”
  - 위 문장에서 컴퓨터 과학을 공부한 것은 누구일까?

Grammar-based approach

Context-free grammars

- Noam Chomsky (1956)에 의해 정의됨
- CFG는 규칙의 집합과 어휘집, 그리고 기호들로 정의된다.
- 종단(terminal) 기호는 단어 혹은 어휘들이다.
- 비종단(non-terminal) 기호는 종단 기호들의 관계적 집합을 가리킨다.
- CFG는 문장의 구조를 분석하거나 구조로부터 문장을 생성하는데 사용될 수 있다.
- 또한 확률모형과도 결합되어 사용될 수 있다.(probabilistic CFG)

Example of CFG (Jurafsky & Martin, 2008): Rules

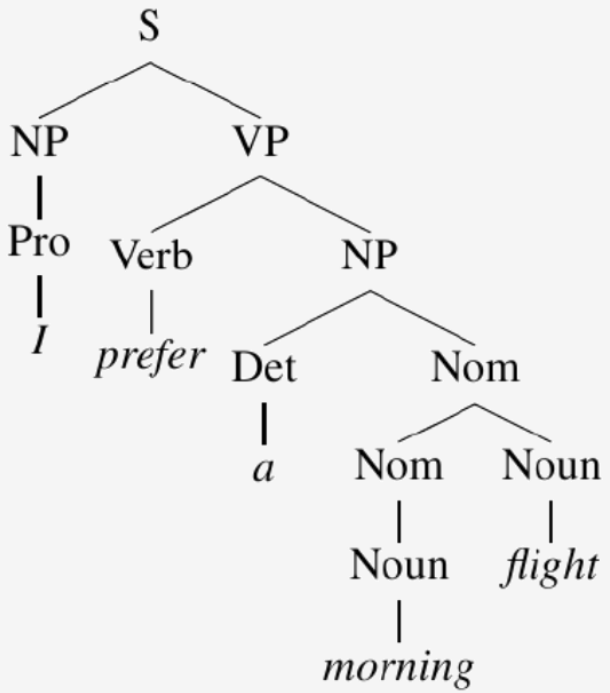
S	$\rightarrow$	*NP VP*	I + want a morning flight
NP	$\rightarrow$	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	$\rightarrow$	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	$\rightarrow$	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	$\rightarrow$	*Preposition NP*	from + Los Angeles

Example of CFG (Jurafsky & Martin, 2008): Lexicon

$$
\begin{aligned}
\textit{Noun} &\rightarrow \textit{flights} \mid \textit{breeze} \mid \textit{trip} \mid \textit{morning} \mid \ldots \\
\textit{Verb} &\rightarrow \textit{is} \mid \textit{prefer} \mid \textit{like} \mid \textit{need} \mid \textit{want} \mid \textit{fly} \\
\textit{Adjective} &\rightarrow \textit{cheapest} \mid \textit{non-stop} \mid \textit{first} \mid \textit{latest} \\
&\quad \mid \textit{other} \mid \textit{direct} \mid \ldots \\
\textit{Pronoun} &\rightarrow \textit{me} \mid \textit{I} \mid \textit{you} \mid \textit{it} \mid \ldots \\
\textit{Proper-Noun} &\rightarrow \textit{Alaska} \mid \textit{Baltimore} \mid \textit{Los Angeles} \\
&\quad \mid \textit{Chicago} \mid \textit{United} \mid \textit{American} \mid \ldots \\
\textit{Determiner} &\rightarrow \textit{the} \mid \textit{a} \mid \textit{an} \mid \textit{this} \mid \textit{these} \mid \textit{that} \mid \ldots \\
\textit{Preposition} &\rightarrow \textit{from} \mid \textit{to} \mid \textit{on} \mid \textit{near} \mid \ldots \\
\textit{Conjunction} &\rightarrow \textit{and} \mid \textit{or} \mid \textit{but} \mid \ldots
\end{aligned}
$$

Example of CFG (Jurafsky & Martin, 2008): Parse Trees

I prefer a morning flight.

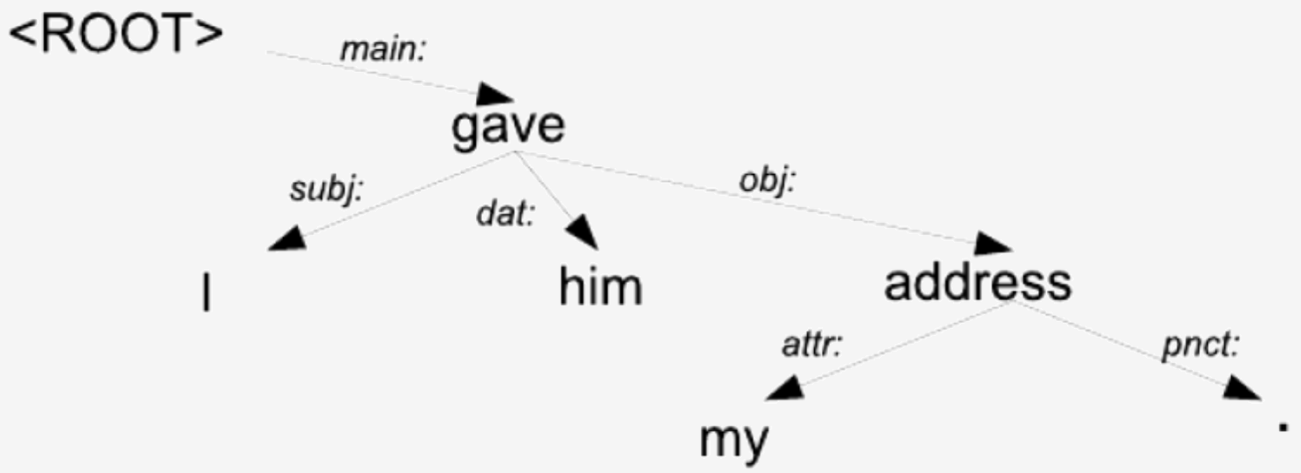


[$_S$ [$_{NP}$ [$_{Pro}$ I]] [$_{VP}$ [$_V$ prefer] [$_{NP}$ [$_{Det}$ a] [$_{Nom}$ [$_N$ morning] [$_{Nom}$ [$_N$ flight]]]]]]

Grammar-based approach

Dependency grammar

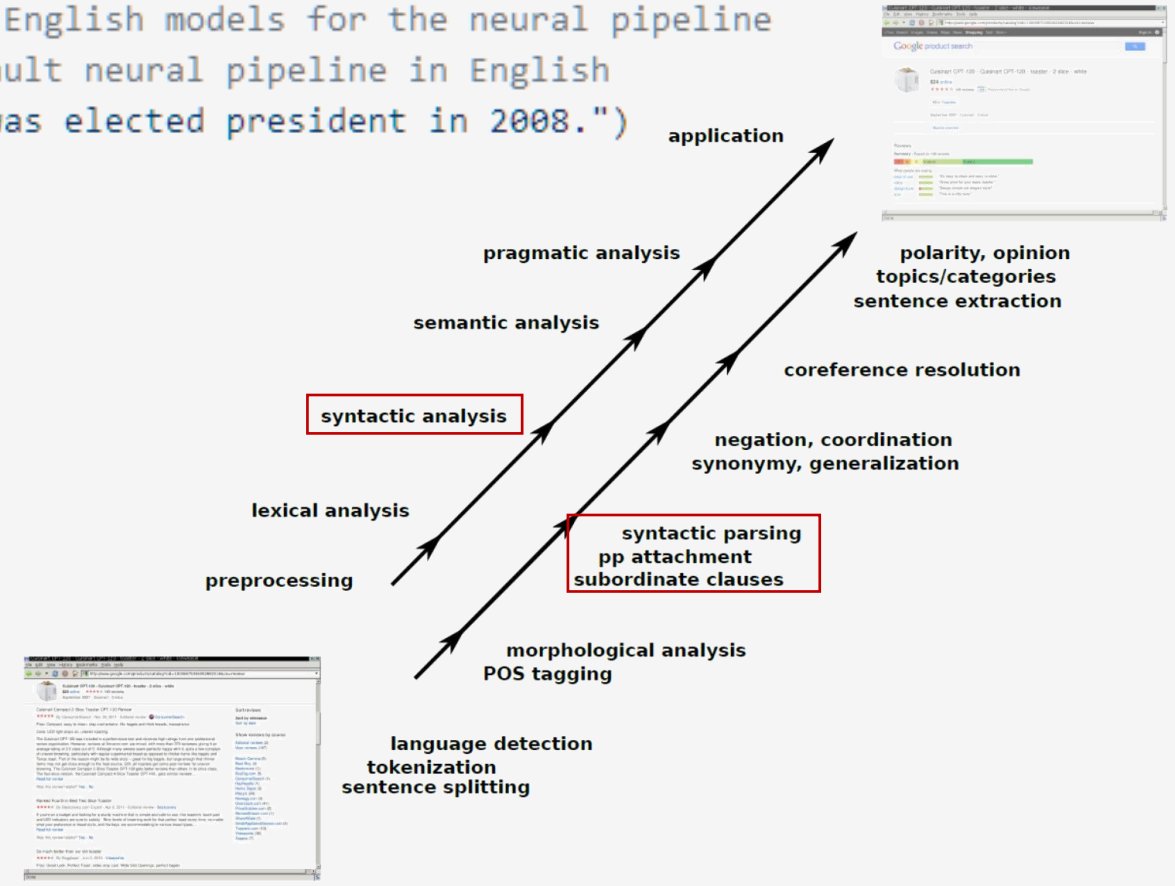
- CFG는 phrase structure grammar를 활용하는 방법이다.
- Dependency grammar는 이와는 달리 단어와 단어간의 의존관계를 활용하는 방법이다.
- 의존성 문법은 한국어처럼 어순이 보다 자유로운 언어들을 기술할때 유리하다.



Application of Dependency Grammar

```
>>> import stanza
>>> stanza.download('en')       # This downloads the English models for the neural pipeline
>>> nlp = stanza.Pipeline('en') # This sets up a default neural pipeline in English
>>> doc = nlp("Barack Obama was born in Hawaii.  He was elected president in 2008.")
>>> doc.sentences[0].print_dependencies()
```

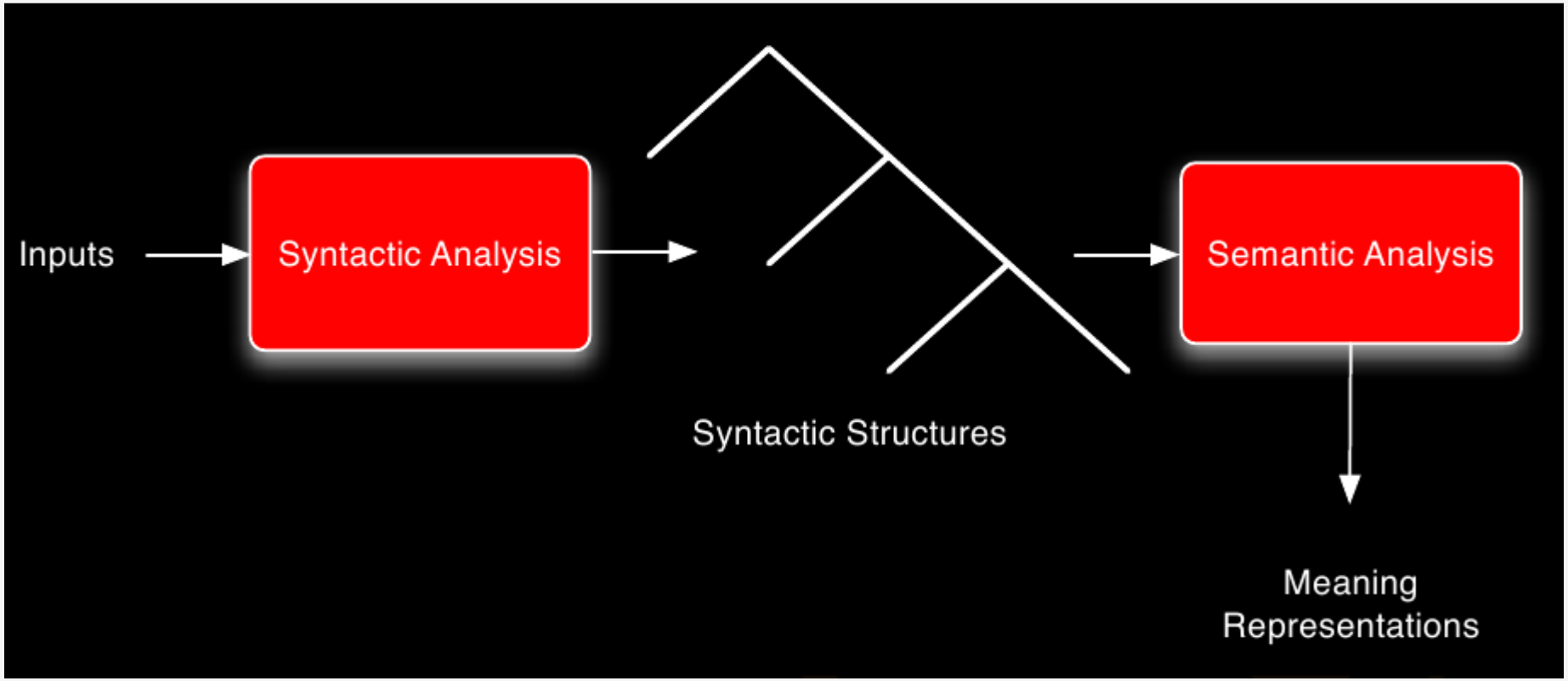
```
('Barack', '4', 'nsubj:pass')
('Obama', '1', 'flat')
('was', '4', 'aux:pass')
('born', '0', 'root')
('in', '6', 'case')
('Hawaii', '4', 'obl')
('.', '4', 'punct')
```



Semantic Analysis

- 언어의 의미를 이해하기 위해서는 언어의 기호와 그 기호가 가리키는 대상을 연결시킬 수 있어야 한다.
- 언어가 가리키는 대상은 언어적이지 않다.
- 통사분석은 이 과제에는 불충분하다.
- 비언어적 세계 지식:
  - 공통 상식
  - 절차적 지식
  - 세계의 대상들과 그 관계들에 대한 지식

Syntax-Semantics Pipeline



Representation of Meaning

의미의 표상은 대상들, 대상들의 속성과 그들의 관계에 대한 기호들로 이루어진다.

의미의 표상은 두 측면에서 바라볼 수 있다.

- 언어적 입력의 표상에서
- 세계의 사태(a state of affairs)의 표상에서

"The student has a notebook."

Possessing:

Possessor: Student

PossessThing: Notebook

$$\exists x, y \;\; Possessing(x) \wedge Possessor(student, x) \wedge PossessThing(y, x) \wedge Notebook(y)$$

Representation of Meaning

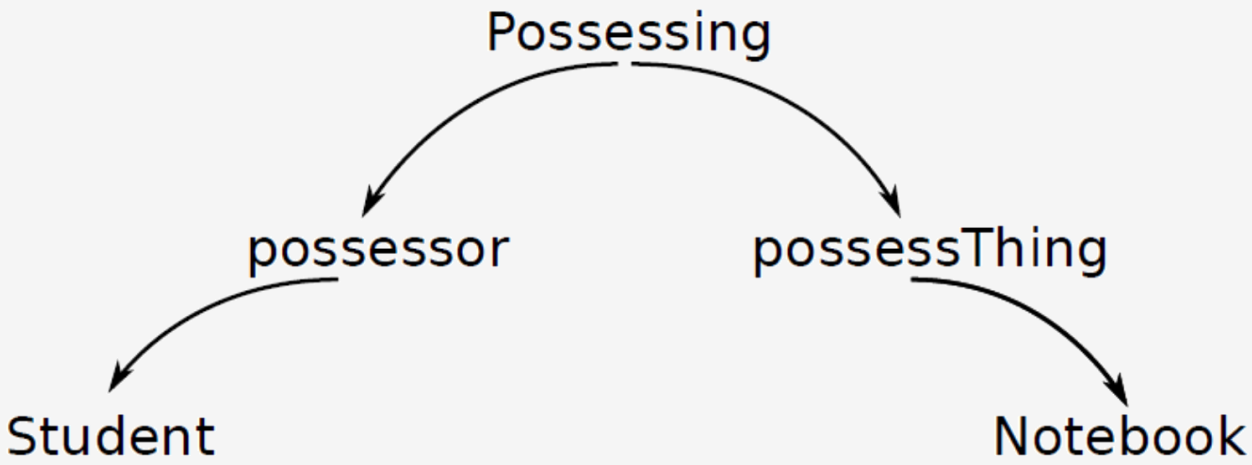
"The student has a notebook."

Possessing:

Possessor: Student

PossessThing: Notebook

$$\exists x, y \;\; Possessing(x) \wedge Possessor(student, x) \wedge PossessThing(y, x) \wedge Notebook(y)$$



Methods of Meaning Representation

- First-order Logic (FOL)
- Description Logic
- Semantic Networks, Frames

First-order Logic (FOL)

First Order Logic (FOL) and First Order Predicate Calculus (FOPC) consist of terms:

- Constants: a constant refers to exactly one object in the world Prof_Kim, SNU
- Functions: functions are single argument predicates; they are also terms in that they refer to unique objects

  LocationOf(SNU)
- Variables: variables allow us to make assertions and to draw inferences

. . . and relations between terms:

- Predicates: refer to relations that hold between objects

  Teaches(Prof_Kim, AI)

  Professor(Prof_Kim) (can be used to express class membership)

FOL example of Jurafsky & Martin (2008)

$$
\begin{aligned}
\textit{Formula} &\rightarrow \textit{AtomicFormula} \\
&\mid \textit{Formula Connective Formula} \\
&\mid \textit{Quantifier Variable}, \ldots \textit{Formula} \\
&\mid \neg \textit{Formula} \\
&\mid (\textit{Formula}) \\
\\
\textit{AtomicFormula} &\rightarrow \textit{Predicate}(\textit{Term}, \ldots) \\
\\
\textit{Term} &\rightarrow \textit{Function}(\textit{Term}, \ldots) \\
&\mid \textit{Constant} \\
&\mid \textit{Variable} \\
\\
\textit{Connective} &\rightarrow \wedge \mid \vee \mid \Rightarrow \\
\textit{Quantifier} &\rightarrow \forall \mid \exists \\
\textit{Constant} &\rightarrow A \mid \textit{VegetarianFood} \mid \textit{Maharani} \cdots \\
\textit{Variable} &\rightarrow x \mid y \mid \cdots \\
\textit{Predicate} &\rightarrow \textit{Serves} \mid \textit{Near} \mid \cdots \\
\textit{Function} &\rightarrow \textit{LocationOf} \mid \textit{CuisineOf} \mid \cdots
\end{aligned}
$$

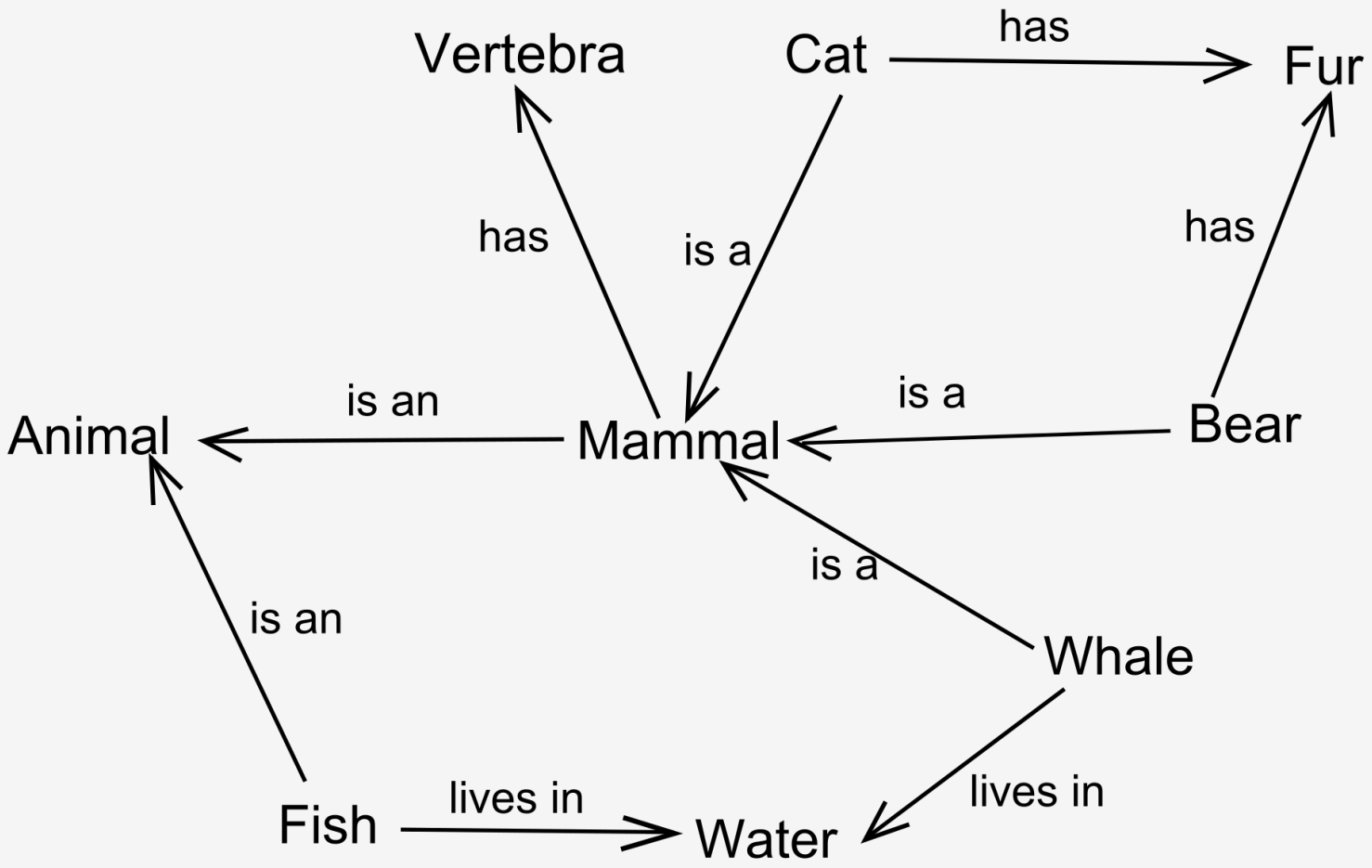
Description Logic

- Subset of FOL
- 시맨틱 네트워크의 이론적 형식화
- 개별자와 범주로 구성되는 온톨로지(ontologies)를 표현하는데 쓰인다.

$$\forall x\ KoreanRestaurant(x) \implies$$
$$Restaurant(x) \wedge (\exists y\ Serves(x, y) \wedge KoreanFood(y))$$

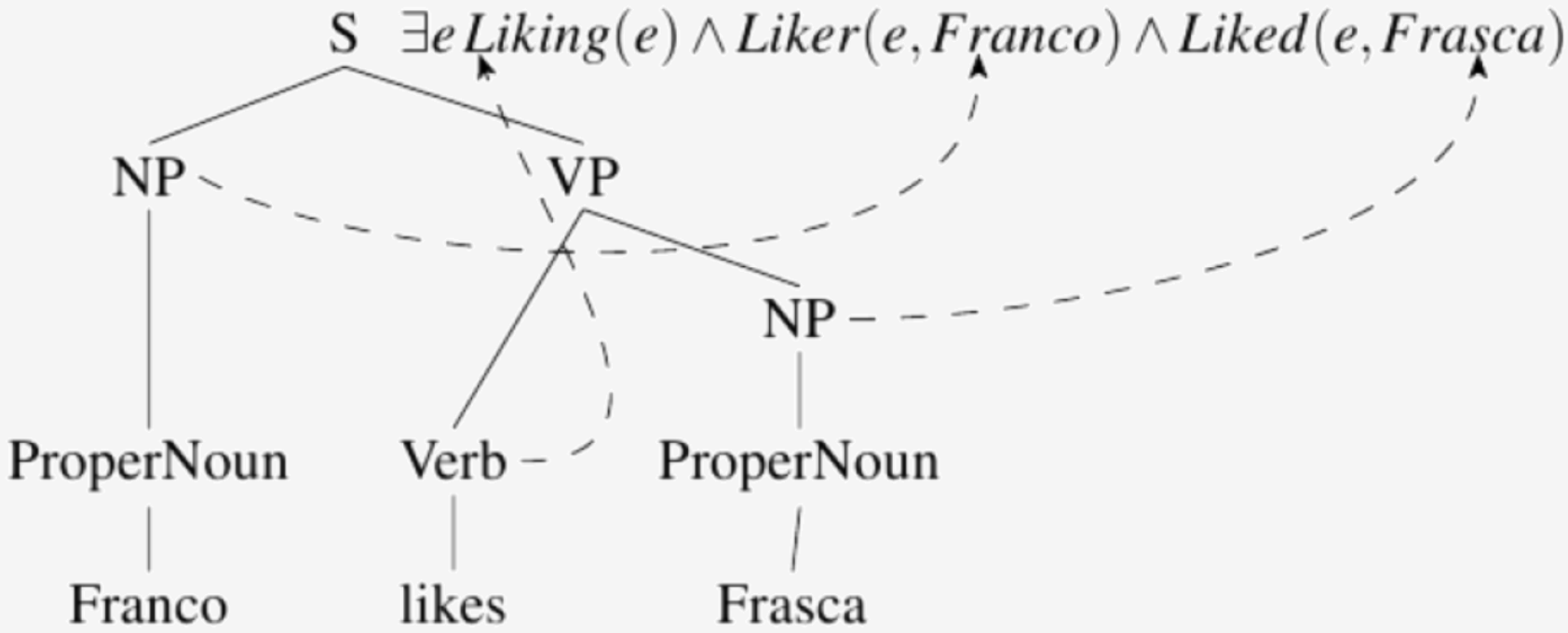
$$KoreanRestaurant \sqsubseteq Restaurant \sqcap \exists hasCuisine.Korean$$

Semantic Networks



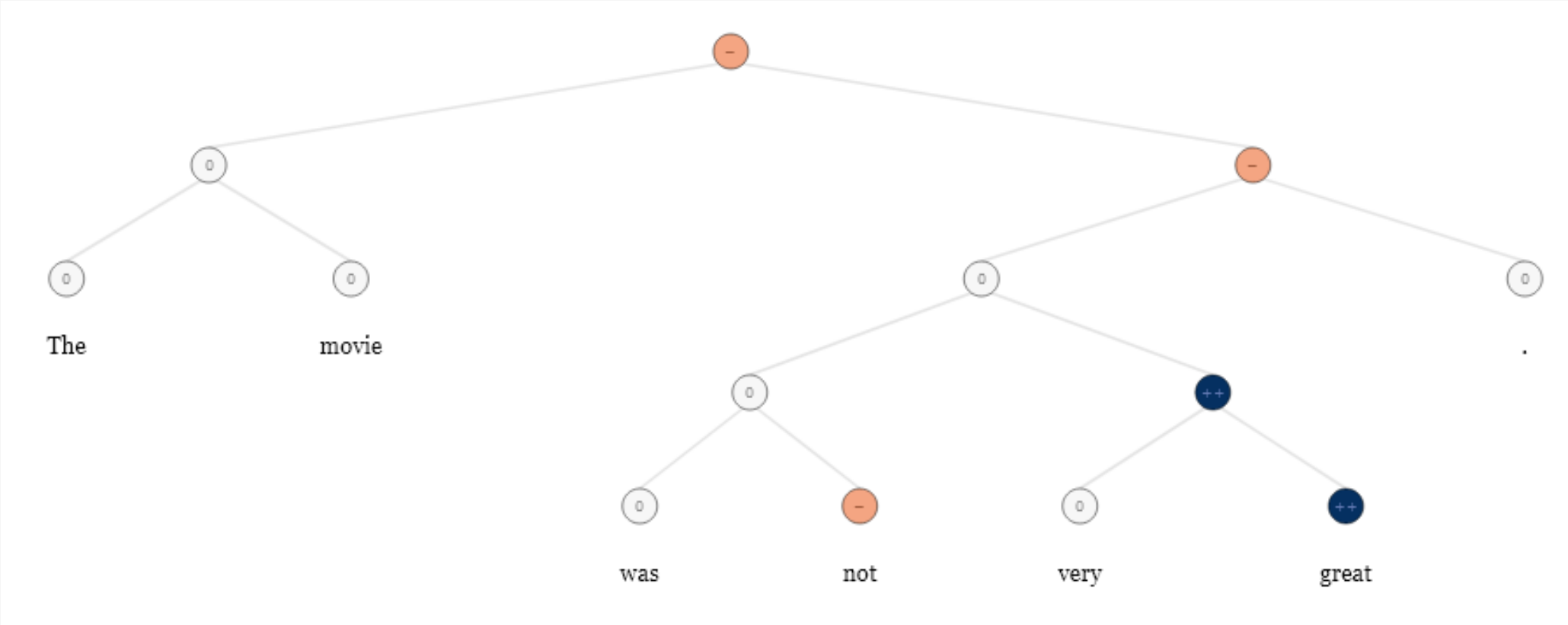
Problem of Semantic Analysis

Compositionality: 문장의 의미는 구성요소들의 의미로부터 구성된다.



Example: sentiment analysis

[Stanford Sentiment Treebank]



Word Senses

- 한 단어의 의미는 맥락에 따라 크게 달라질 수 있다.
  - Bank: 금융기관 vs. 강가 vs. 저장소
  - Bright: 밝음 vs. 똑똑한
  - 정의: 사회적 [정의] vs. 사전적 [정의]
  - 잠자리: 잠자는 곳 vs. 곤충
- 한국어는 한자표기는 다르지만 발음은 같은 탓에서 발생한 동음이의어가 많다.
- 단어의 의미 유형
  - 동음이의어, 다의어, 환유
- 단어들 간의 의미 유형
  - 유의 관계, 반의 관계, 부분 관계

Word Sense Disambiguation

- 단어 의미의 다의성은 대부분의 NLP 문제에 있어 골치거리이다.
- 단어 사전 자료: WordNet
- 입력: 텍스트의 단어들
- 출력: 단어 의미가 태깅된 텍스트

- WordNet 사용예제

```
>>> from nltk.corpus import wordnet
>>> w1 = wordnet.synset('ship.n.01' )
>>> w2 = wordnet.synset('boat.n.01' )
>>> print(w1.wup_similarity(w2))
```

```
0.9090909090909091
```

Semantic Role Labeling (SRL)

- 문장의 각 술어에 대한 의미역(semantic role)을 자동으로 판별하는 과제
- 흔히 PropBank와 같은 labeled data를 지도학습을 통해 학습시키는 방법을 쓴다.
- PropBank

Frameset **agree.01**

Arg0:	Agreer
Arg1:	Proposition
Arg2:	Other entity agreeing
Ex1:	[$_{Arg0}$ The group] *agreed* [$_{Arg1}$ it wouldn't make an offer unless it had Georgia Gulf's consent].
Ex2:	[$_{ArgM-Tmp}$ Usually] [$_{Arg0}$ John] *agrees* [$_{Arg2}$ with Mary] [$_{Arg1}$ on everything].

fall.01 "move downward"

Arg1:	Logical subject, patient, thing falling
Arg2:	Extent, amount fallen
Arg3:	start point
Arg4:	end point, end state of arg1
ArgM-LOC:	medium
Ex1:	[$_{Arg1}$ Sales] *fell* [$_{Arg4}$ to $251.2 million] [$_{Arg3}$ from $278.7 million].
Ex1:	[$_{Arg1}$ The average junk bond] *fell* [$_{Arg2}$ by 4.2%] [$_{ArgM-TMP}$ in October].

Coreference Resolution

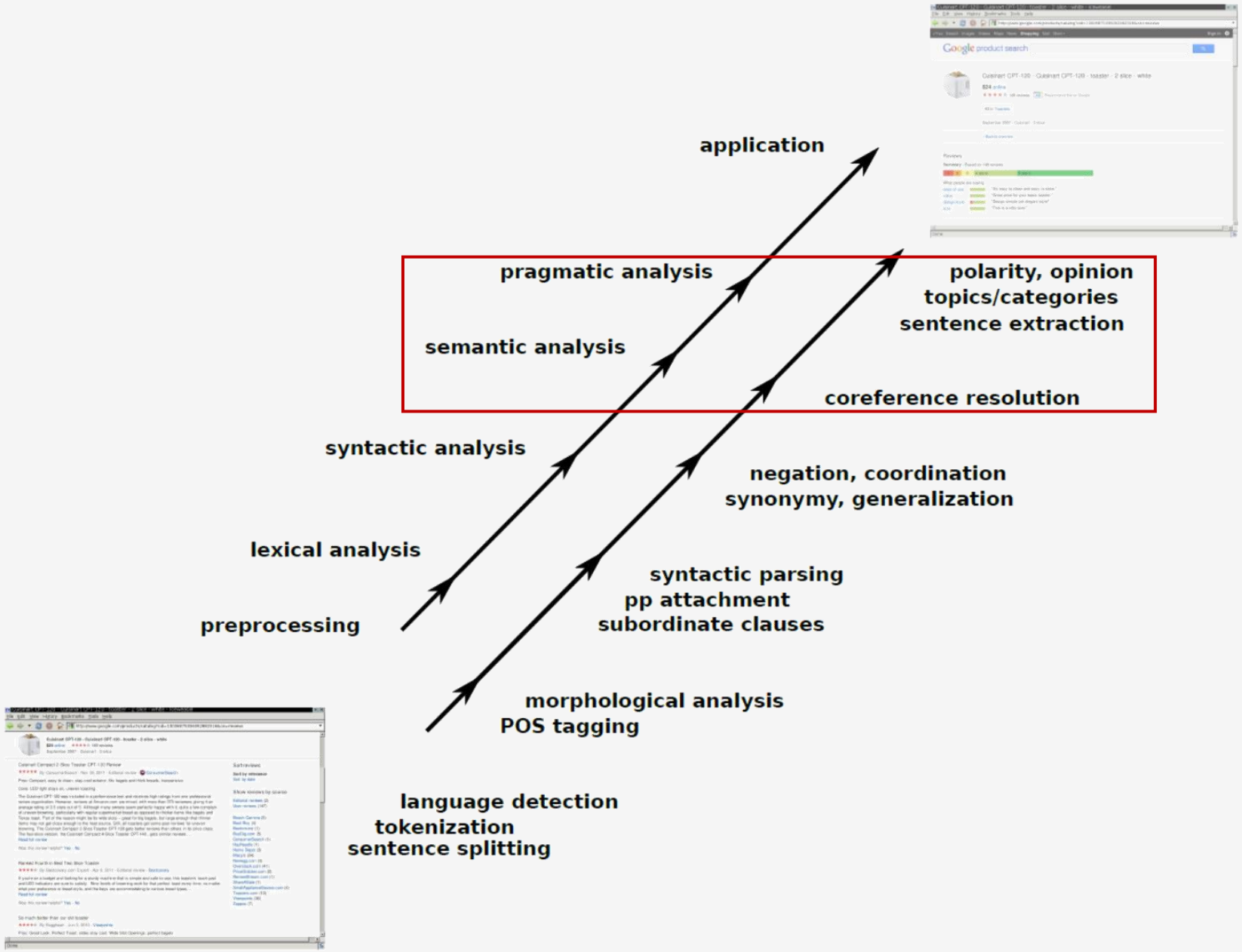
- 문장 혹은 문단에서 언급된 같은 대상들의 동일 지시성을 판별하는 과제
- 용어들:
  - Mentions: 어떤 entity를 가리키는 언어 표현
  - Entities (referents): 지시된 대상 자체
  - Coreference: 둘 이상의 mention이 같은 대상을 지시할때 그것을 공통 지시적이라고 말한다.
  - Anaphora: 앞에서 언급된 대상에 대한 지시표현을 말한다.

Mention Types

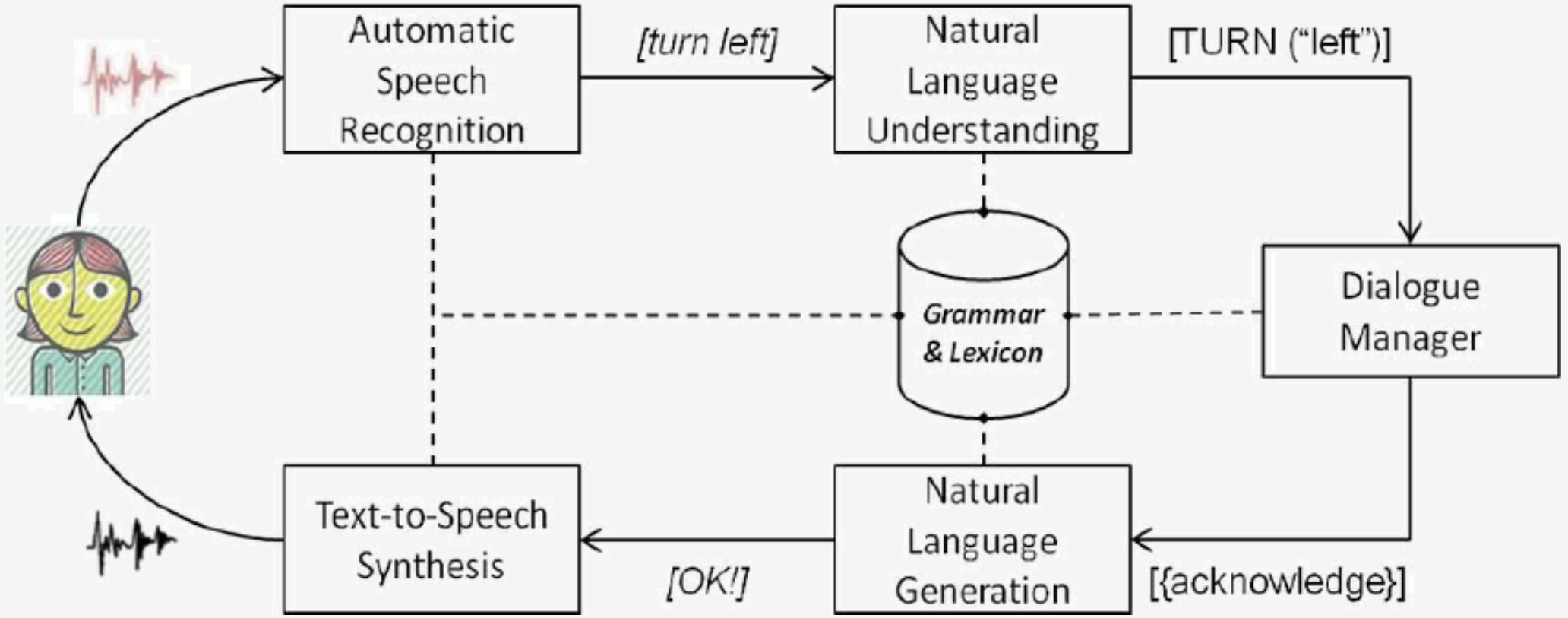
- Indefinite noun phrases: a guy
- Definite noun phrases: the man
- Proper names: John, South Korea, the United States
- Pronouns
  - Personal pronouns: I, you, he, she, it
  - Possessive pronouns: mine, yours, his, her
  - Reflexive Pronouns: myself, yourself, himself
  - Demonstrative pronouns: that, these, those

Tools

- Annotated Corpora
  - MUC-6, MUC-7 (Message Understanding Conference) – through LDC
  - ACE 2002 - ACE 2005 (Automatic Content Extraction) – through LDC
  - OntoNotes 4.0 (CoNLL Shared Task 2011), OntoNotes 5.0 (CoNNL Shared Task 2012 – Arabic, Chinese, English) – through LDC
  - i2b2/VA 2011 (medical domain)
  - SemEval 2010 (Catalan, Dutch, English, German, Italian, Spanish) – through LDC
  - Genia (documents about molecular biology)
- Open sourced system
  - Stanford system: https://nlp.stanford.edu/software/dcoref.shtml


application
pragmatic analysis
semantic analysis
polarity, opinion
topics/categories
sentence extraction
coreference resolution
syntactic analysis
negation, coordination
synonymy, generalization
lexical analysis
syntactic parsing
pp attachment
subordinate clauses
preprocessing
morphological analysis
POS tagging
language detection
tokenization
sentence splitting

Dialogue Systems: Final Art



Brief History of NLP

"사람들 사이에 이견이 생기게 되면 누군가 '계산합시다'라고 말하게 될 것이고, 그럼 누가 옳은지 바로 알 수 있게 될 것이다."

Leibniz (1646-1716)

원초적 NLP 도구: 논리학

삼단논법

- 전제
  - 모든 사람은 죽는다.
  - 소크라테스는 사람이다.
- 결론
  - 소크라테스는 죽는다.

아리스토텔레스의 삼단논법은 언어적 사고의 체계화를 향한 시도라고 볼 수 있다.

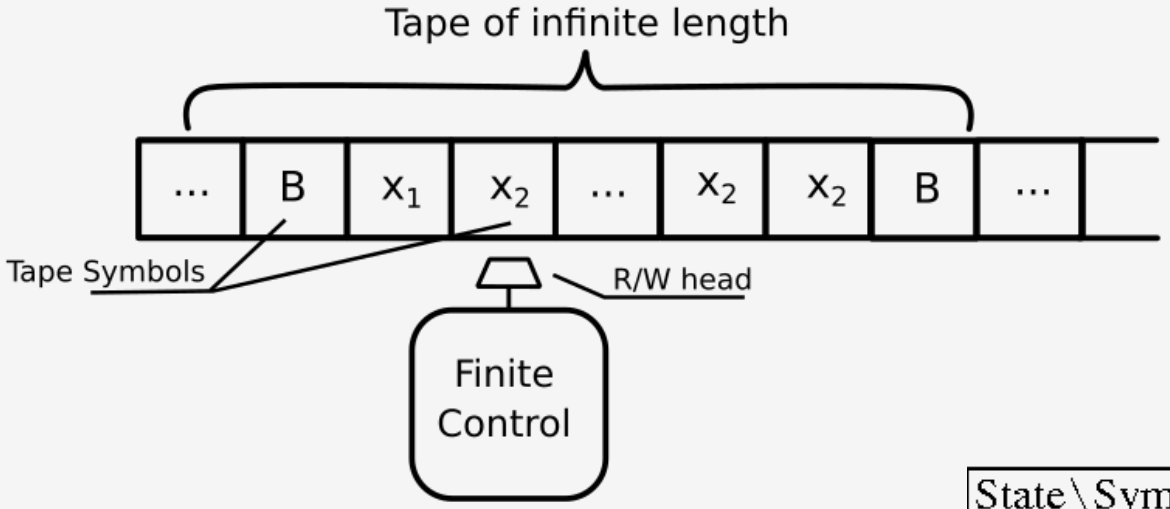
논리학의 진화

- 수학에 비해 논리학의 발전은 느렸다.
- 라이프니츠는 보편 기호체계를 제안했으나 실제적 이론화에는 실패했다.
- 1854년 영국의 수학자 조지 불이 불 대수(Boolean algebra)를 발표
  - 대수적 계산을 통해 삼단추론의 형식적 제약 및 약점을 극복
  - 논리학의 수학적 모형을 제시
- 1879년 독일의 수학자 프레게가 명제 논리학을 발표
  - 기호 논리학, 논리주의의 시작
  - 조건문 기호(→) 및 양화사(∀, ∃)가 사용됨
- 1910년 영국의 러셀과 화이트헤드가 <수학원리>를 출판
  - 1차 술어 논리학(FOL)이 처음으로 등장함

보편기계의 등장

- 1929년 독일의 수학자 괴델은 1차 술어논리학의 완전성을 증명한다.
- 1930년 괴델은 이어서 산술체계의 불완전성을 증명하면서 괴델 기수법을 소개한다.
  - 괴델 기수법은 1차 술어 논리식을 자연수와 일대일로 맵핑을 시켜주는 기법이었다.
  - 괴델 기수법은 산술 계산을 통해서 복잡한 논리식을 기계적으로 처리할 수 있다는 것을 보여준 것이었다.
  - 이는 전산적 인코딩 개념이 탄생했음을 의미한다.
- 1937년 영국의 앨런 튜링이 괴델 기수법에 영감을 얻어 튜링기계에 대한 논문을 발표한다.
  - 현대적 컴퓨터의 이론적 모형이 발표됨
  - 튜링기계는 상태(state)와 읽고 쓰기가 가능한 head, 행동표를 통해 형식적으로 기술 가능한 모든 행동을 따라할 수 있다.
  - 이 기계는 모든 기계를 흉내낼 수 있는 기계, 즉 보편기계를 의미했다.

Turing Machine

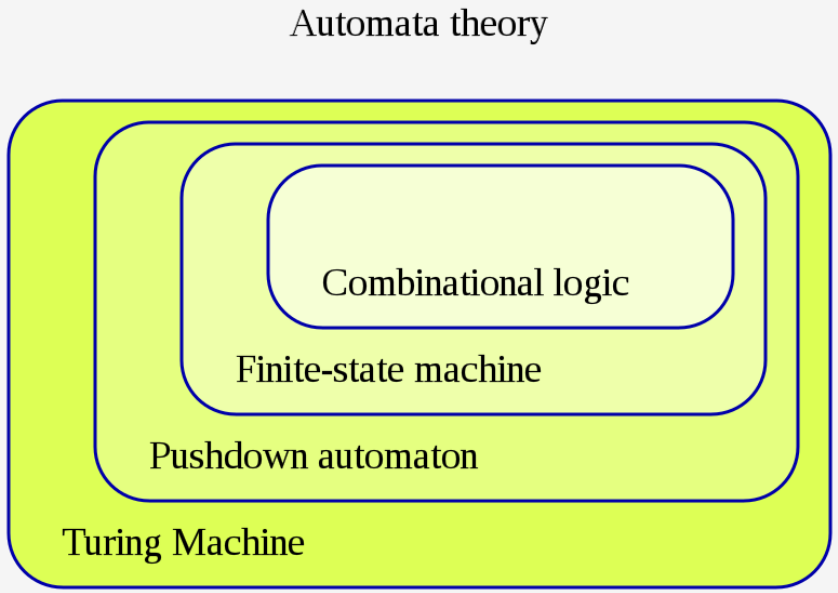


Behavior Table

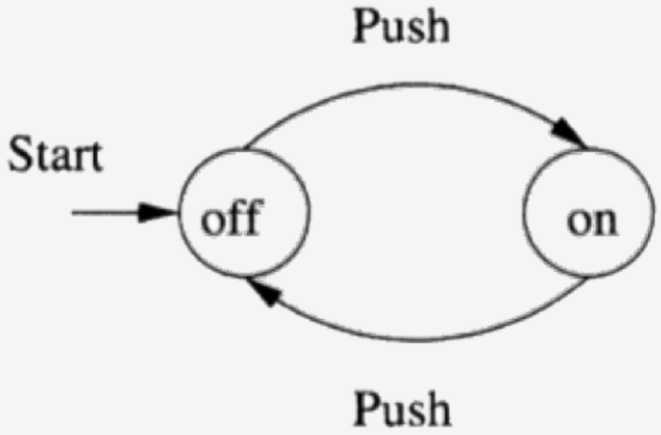
State\Symbol	0	1	X	Y	B
s0	(s1,X,R)	–	–	(s3,Y,R)	–
s1	(s1,0,R)	(s2,Y,L)	–	(s1,Y,R)	–
s2	(s2,0,L)	–	(s0,X,R)	(s2,Y,L)	–
s3	–	–	–	(s3,Y,L)	(s4,B,R)
s4	–	–	–	–	–

Formal Language Theory

- 튜링의 계산이론 모형은 동시대 많은 연구자들에게 깊은 영감을 주었다.
- Chomsky는 튜링기계를 통해 문법을 정의할 수 있음을 깨달았다.
  - 오토마타 이론과 깊게 연관
  - 오토마타의 특정 유형은 특정 유형의 문법과 대응된다.
  - 유한상태 오토마타는 정규표현 문법을 의미한다.
  - Push-down 오토마타는 Context-free grammar를 의미한다.
  - 이를 통해 Chomsky는 형식언어 계산이론으로 자연어를 연구하는 길을 제시했다.

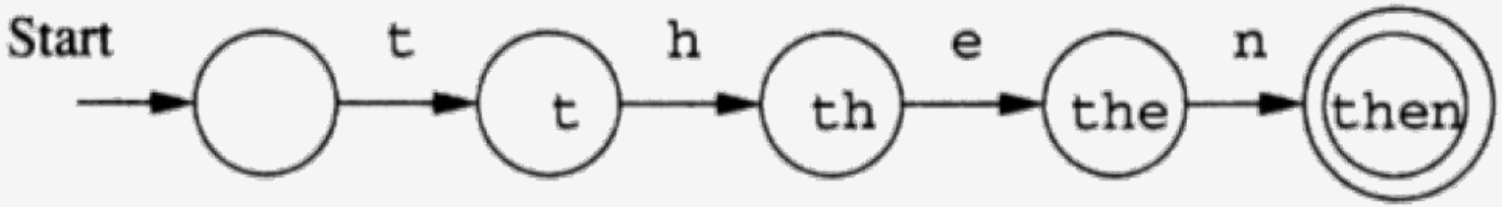


오토마타: 단순한 예제 1



State\Input	Push
On	Left
Off	Right

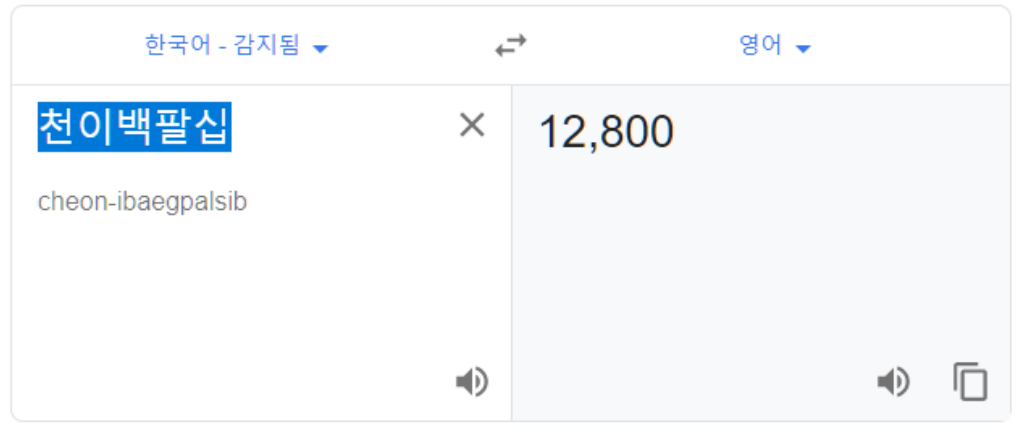
오토마타: 단순한 예제 2



	t	h	e	n
→START	t			
t		th		
th			the	
the				then
*then				

예제: 한국어 숫자표현의 규칙성

- 한국어의 숫자 표기는 매우 규칙적이다.
- 한국어로 작성된 숫자표현들: “일천”, “일백만”, “일천백”
- 한국 숫자표기의 올바른 변환 예:
  - ‘일천’→1000
- 이 과제는 규칙기반 학습을 통해서는 매우 효율적으로 처리가 가능하지만 지도학습으로는 매우 비효율적이다.



한국어 숫자 표기법

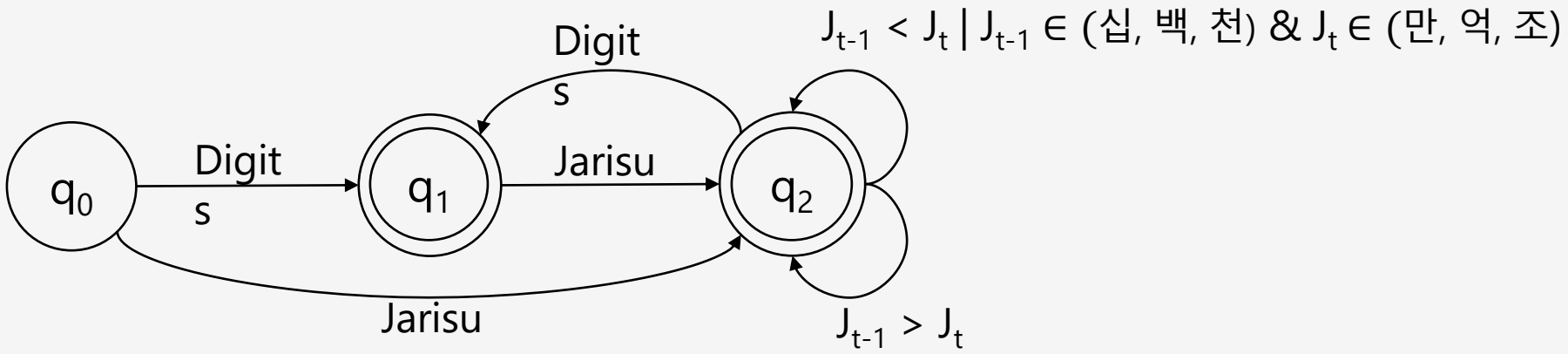
- Digits = {영, 일, 이, 삼, 사, 오, 육, 칠, 팔, 구}
- Jarisu = {십(10^1), 백(10^2), 천(10^3), 만(10^4), 억(10^8), 조(10^{12}), …}

- 삼천일백 = 삼×천(10^3) +일×백(10^2)
- 삼천일백만=(삼×천(10^3) + 일×백(10^2)) × 만(10^4)

한국어에서는 십만 이상부터는 이전의 자릿수를 재사용하는 방법을 쓴다.

- 이십만 = 이(digit) × 십(jarisu) × 만(jarisu) = $2 \times 10^1 \times 10^4$
- 이백만 = 이(digit) × 백(jarisu) × 만(jarisu) = $2 \times 10^2 \times 10^4$

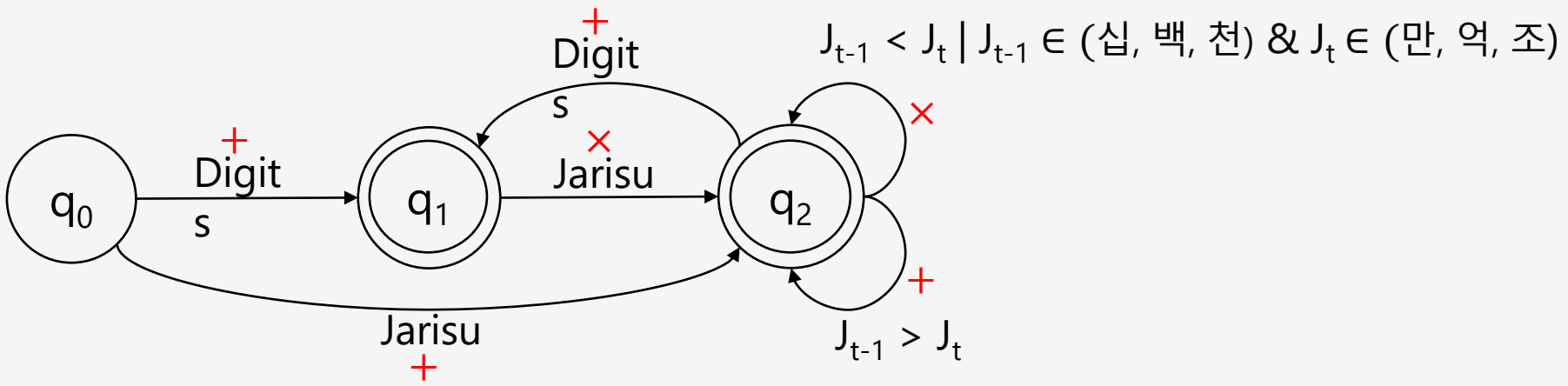
- Digits = {영, 일, 이, 삼, 사, 오, 육, 칠, 팔, 구}
- Jarisu = {십(10^1), 백(10^2), 천(10^3), 만(10^4), 억(10^8), 조(10^{12}), …}



- 삼천일백
- 삼천일백만

- 일백천
- 억만천

- Digits = {영, 일, 이, 삼, 사, 오, 육, 칠, 팔, 구}
- Jarisu = {십(10^1), 백(10^2), 천(10^3), 만(10^4), 억(10^8), 조(10^{12}), …}



- 삼천일백
- 삼천일백만

- 일백천
- 억만천

여러 모형들의 출현: 1940년대와 1950년대

- 초기 언어모형
  - 오토마타 기반 모형
  - 확률 및 정보이론 기반 모형
- 파생 모형들의 출현
  - 퍼셉트론
  - 유한상태 오토마타, 정규식
  - 마르코프 프로세스
- 섀논의 정보이론이 언어모형에 적용됨
  - N-그램 모델
  - 혼잡도(perplexity)

The Two Camps: 1957-1970

- 상징기반 처리 VS 확률기반 처리
- 상징기반 처리
  - 촘스키류의 생성문법
  - 다이내믹 프로그래밍 기반 접근법
- 인공지능 진영
  - 인공신경망 접근
  - 추론과 논리 기반(predicate calculus) 접근법
- 초창기 NLU 모델이 QA 시스템을 위해 등장

Various Paradigms: 1970-1983

- 브라운 코퍼스 등장, 코퍼스 기반 언어처리 등장
- 확률기반 방법이 음성인식 분야에서 크게 발전하기 시작
- 논리기반 진영에서 프롤로그와 같은 함수형 언어 개발
- NLU 분야에서 개념적인 지식에 대한 계산모형의 발전: 스크립트, 프레임, 플래닝, 시맨틱 네트워크
- 담화 모델링 역시 연구가 시작됨. 담화구조 표상, reference resolution 및 화행에 대한 논리기반 처리연구 시작

Empiricism and Finite-State Models Redux:1983-1993

- 유한상태 문법 이론의 재각광
  - 형태론, 품사태깅에 효용성 입증
- IBM 토마스 왓슨 연구소의 확률적 음성인식 모형의 성공
  - Data-driven 방법의 성공
  - 다른 영역으로 활발히 확대됨

The Field Comes Together: 1994-1999

- Data-driven 방법과 확률기반 모형이 대세를 굳히기 시작함
- POS-taggig, dependency parsing, reference resolution, 담화처리 등 모든 영역에서 확률적 학습모형을 채택하기 시작함.
- 이 배경에는 데이터의 축적과 함께 컴퓨터 하드웨어의 빠른 발전이 있었다.

The Rise of Machine Learning: 2000-present

- 많은 종류의 언어 데이터들이 만들어짐
  - 인터넷 시대의 개막과 함께 대용량의 언어자료가 쉽게 접근가능하게 됨
  - Penn Treebank, Prague Dependency Treebank, Porpbank, Penn Discourse Treebank, Timebank etc.
- 많은 데이터들로 인해 지도 학습 알고리즘들이 본격적으로 힘을 발휘하기 시작함
  - SVM, Logistic regression, Graphical Bayesian models, Neural network models

The Reign of Deep Learning: 2006-present

- 이전까지는 다양한 기계학습 방법론 중 하나였던 인공 신경망 방법이 비약적으로 발전
  - 하드웨어적 요인: GPU 활용법 탄생 (Nvidia CUDA)
  - 소프트웨어적 요인: 신경망 아키텍쳐 및 최적화 기술의 발달
- 비지도(unsupervised) 방법들도 점차 주요 방법론으로 떠오름
  - Pretrained embedding methods
  - Autoregressive, VAE methods
  - Transfer learning, self-supervised learning methods
- 심층 강화학습(deep reinforcement learning) 방법도 등장
  - Q&A 시스템이나 대화처리 시스템 연구에 응용됨

NLP의 현재: 딥러닝 전성시대

"한때 인공신경망으로 연구를 하면 논문을 받아주지 않던 시절이 있었는데, 이제는 인공신경망으로 연구를 안하면 논문을 받아주지 않는다."

미상의 연구자

언어모형 (Language Models)

- 어떤 언어적 표현이 나타날지에 대한 예측과제를 생각해보자.
  - "그 사람은 [ ] 좋아한다."
- 이런 예측은 여러 범주의 지식에 의해 결정된다.
  - 대화주제, 문법지식, 어휘지식 등
- 언어 자료의 통계적 일반화는 이 복잡성에 대해 단순한 해법을 제공한다.
- 언어적 요소들의 확률분포 = 언어모형
- 성공적인 언어모형은 어떤 언어 표현이 나타날 확률을 정확하게 계산할 수 있어야 한다.
  - $\mathrm{P}(\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_\mathrm{n}) \in \{0,1\}$

N-Gram Models: 고전적 언어모형

- N-Gram 이란 단어들의 연쇄 단위를 가리키며 앞의 N은 연쇄 단위의 크기를 뜻한다.
- Unigrams, Bigrams, Trigrams, …, N-grams
- e.g., Bigrams:
  - S = "나는 어저께 공부를 했다"
  - Bigrams(S) = (나는, 어저께), (어저께, 공부를), (공부를, 했다)
- N-Gram 모형은 한 단어의 확률을 그 이전 단어 연쇄열과의 조건부 확률을 통해 예측한다.
  - P(했다|나는 어저께 공부를) → P(했다|(나는, 어저께), (어저께, 공부를))
- 최고의 N-Gram 모형
  - Katz's back-off trigram model

Bengio et al (2003)의 비판

- N-Gram 모형이 제공하는 언어모형은 의미적 차원의 예측력이 너무 약하다.
- “The cat is walking on the street”를 학습한 언어모형은 “The cow is running down the street”와 같은 문장이 또한 나타날 수 있음을 예측할 수 있어야 한다.
- 하지만 학습데이터에서 ‘cow is running’, ‘running down the’, ‘down the street’와 같은 표현들이 적었다면 N-Gram 모형은 이 문장의 출현확률을 실제확률보다 낮게 잡을 수 밖에 없다.
- 단어의 입력을 단순히 문자열로만 보아서는 이 문제를 해결할 수 없다.

Bengio et al (2003)의 대안

- 단어 벡터 모형의 도입
  - 기존 연구(Schutze, 1992)와 같이 단어를 벡터공간에 임베딩 시킨다.
  - 단어 임베딩 벡터를 인공신경망의 입력벡터로 이용한다.

- 다층 신경망 예측모형을 언어모형에 적용
  - 인공신경망은 지도학습을 통해 두 가지 임무를 동시에 수행한다.

    1) 단어 임베딩 벡터 학습

    2) 주어진 맥락 단어 다음에 올 단어를 예측하기

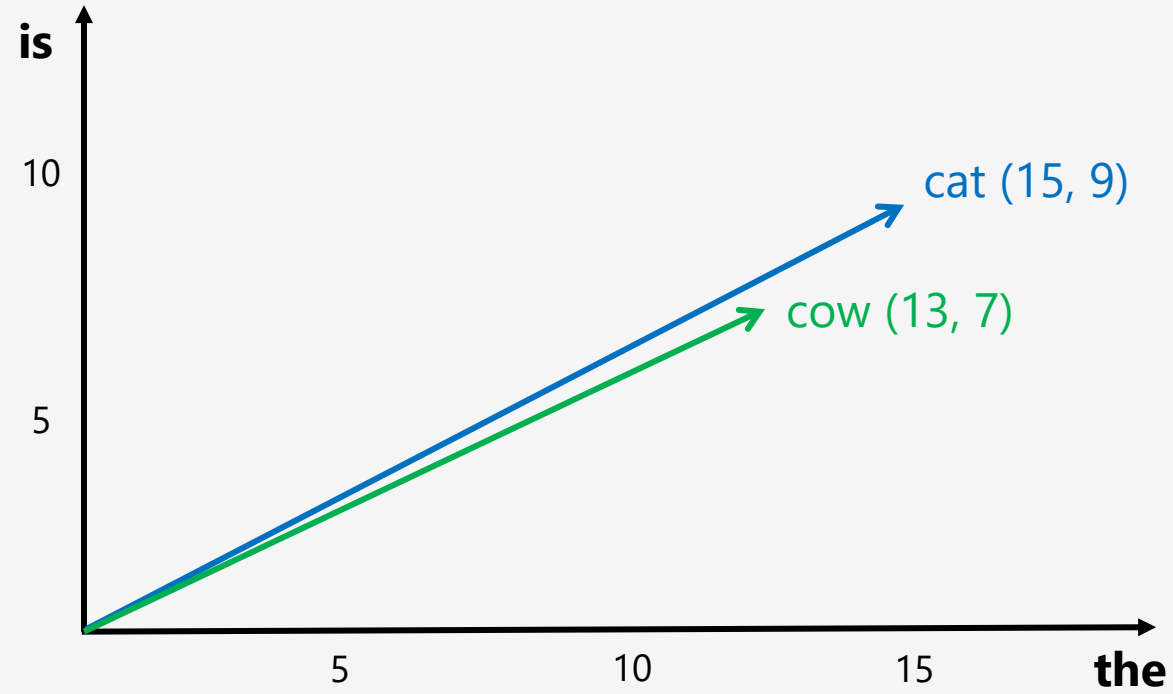
Vector Space Models of Words (word embeddings)

Word-Context matrix

	C_1	C_2	C_3
W_1	-	1	0
W_2	1	-	1
W_3	0	1	-

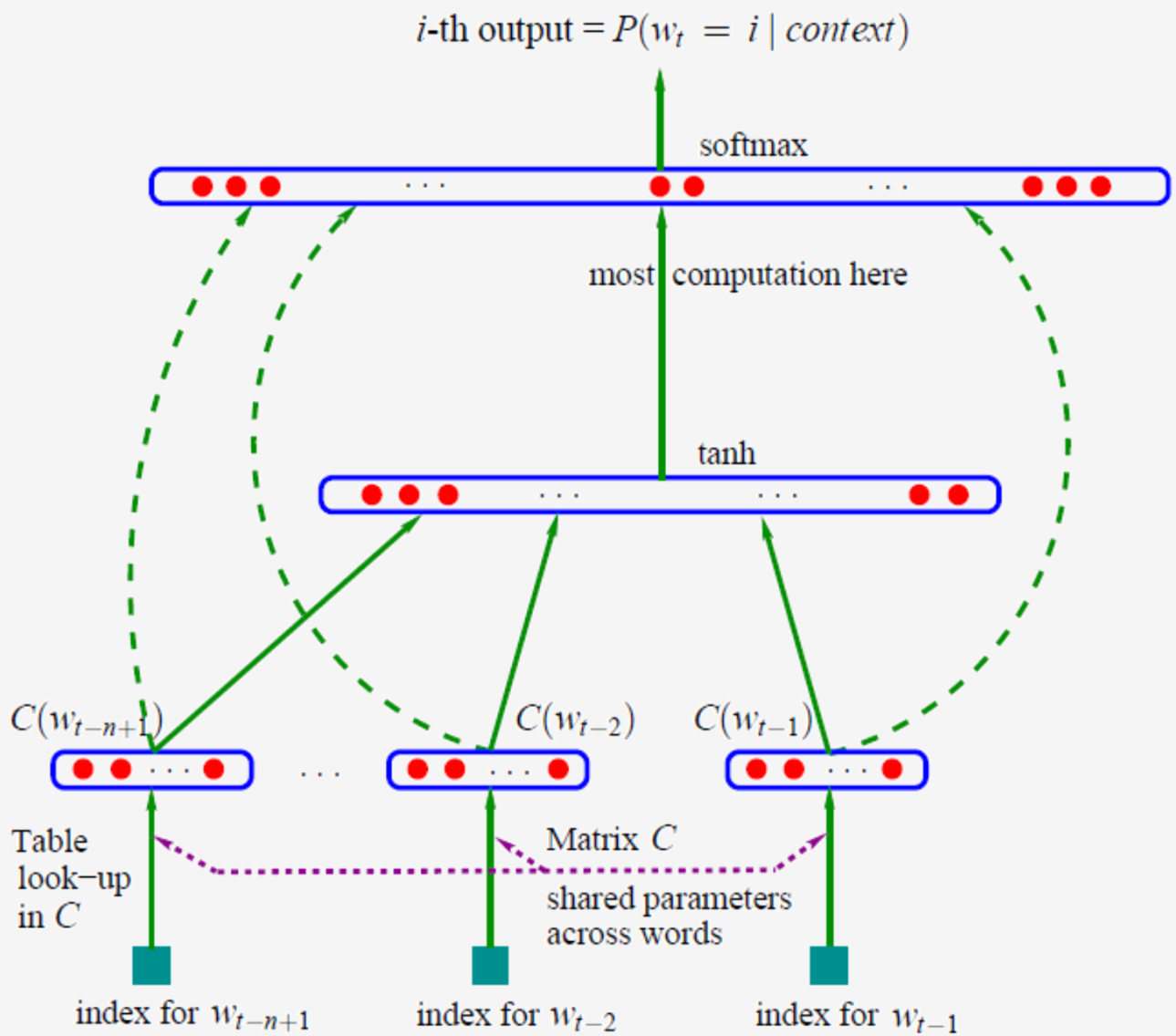
- 벡터 공간 모형(Vector Space Models)은 Distributional hypothesis와 깊이 연관된 모형이자 방법론이다.
- 맥락 벡터(context vectors)를 통해 특정 어휘를 벡터 공간으로 사상한다.
- 어휘의 벡터를 통해 어휘 간의 의미적 유사성을 계산하거나 공간적 표상을 수행할 수 있다.

word\context	the	cat	cow	Is	running
the	100	15	13	10	8
cat	15	100	0	9	6
cow	13	0	100	7	1
is	10	9	7	100	35
running	8	6	1	35	100



Neural Probabilistic Language Model

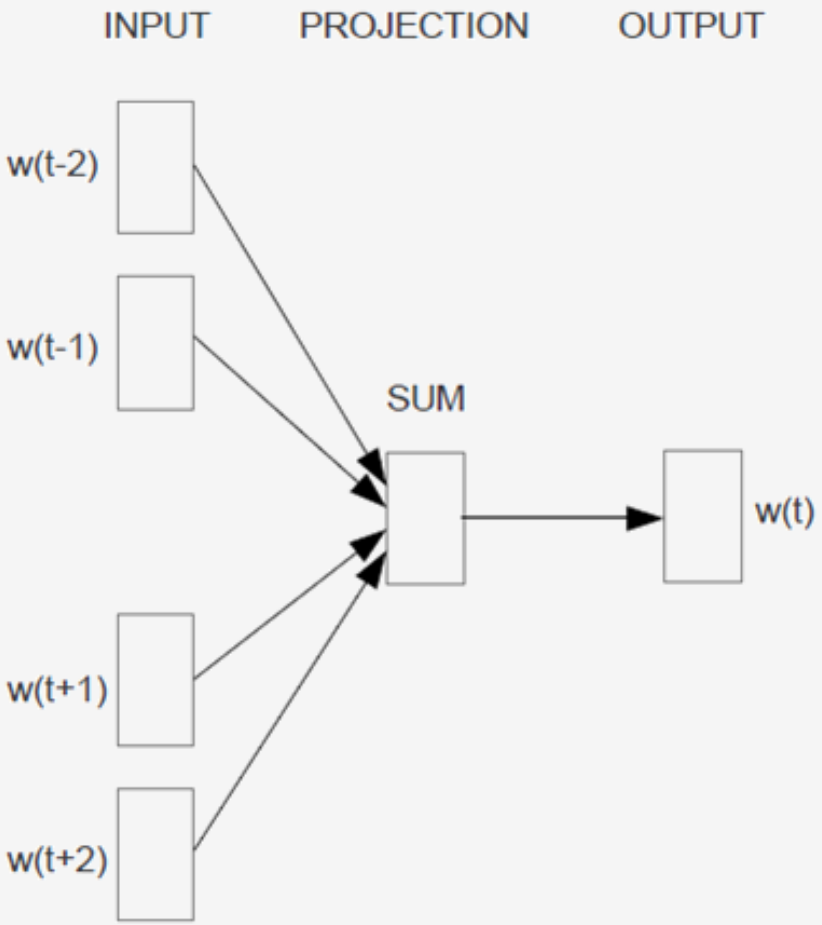
- Bengio et al (2003)은 다음의 중요한 개념들을 제시함
  - 단어 문자열이 아니라 단어 임베딩 벡터를 입력으로 사용
  - 단어 벡터를 연결한 입력에 대한 인공신경망의 계산을 공통 확률함수로 볼 수 있음
  - 단어 임베딩 벡터는 신경망과 함께 동시에 학습되거나 사전 학습된(pretrained) 벡터를 통해 초기화시켜 사용될 수 있음
- 이후 위 개념들은 NLP 분야에서 표준적 방법론으로 자리잡게 됨
- 이때 Bengio가 제시한 것은 다소 단순한 feedforward neural network였다.
- 당시에는 GPU를 활용할 수 없었고, Bengio는 매우 복잡한 방식으로 CPU 클러스터를 이용해 병렬계산을 해야 했다.
- Bengio의 연구는 이후 Word2Vec, GLoVe와 같은 임베딩 모형의 출현에 영향을 주었다.


i-th output = $P(w_t = i \mid context)$
softmax
most computation here
tanh
$C(w_{t-n+1})$
$C(w_{t-2})$
$C(w_{t-1})$
Table look–up in C
Matrix C
shared parameters across words
index for w_{t-n+1}
index for w_{t-2}
index for w_{t-1}

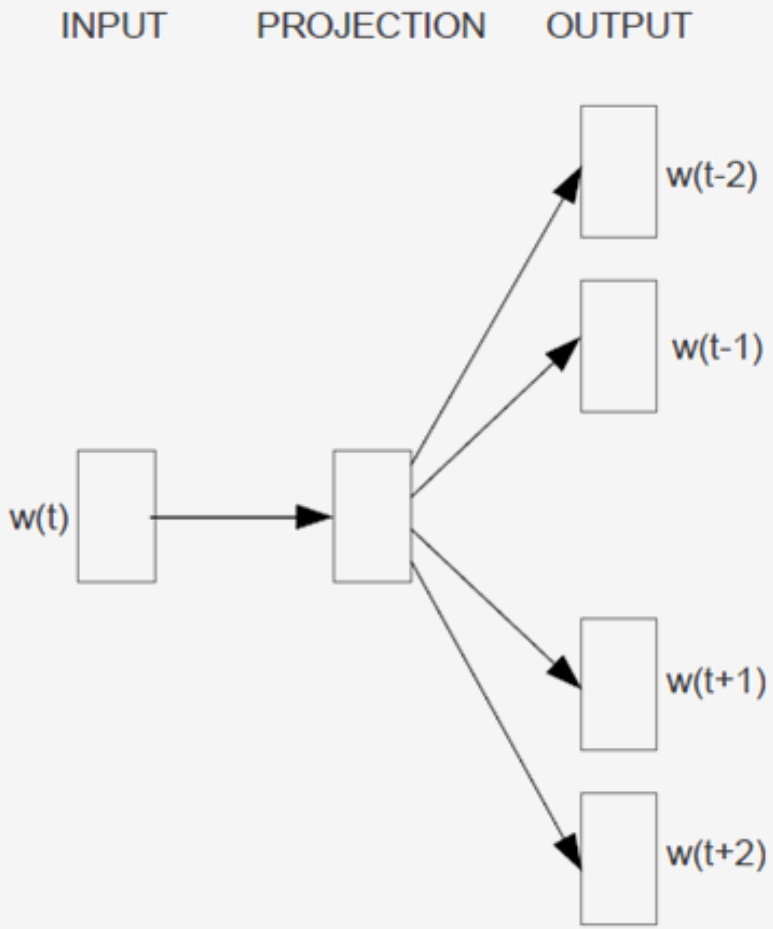
	n	c	h	m	direct	mix	train.	valid.	test.
MLP1	5		50	60	yes	no	182	284	268
MLP2	5		50	60	yes	yes		275	257
MLP3	5		0	60	yes	no	201	327	310
MLP4	5		0	60	yes	yes		286	272
MLP5	5		50	30	yes	no	209	296	279
MLP6	5		50	30	yes	yes		273	259
MLP7	3		50	30	yes	no	210	309	293
MLP8	3		50	30	yes	yes		284	270
MLP9	5		100	30	no	no	175	280	276
MLP10	5		100	30	no	yes		265	**252**
Del. Int.	3						31	352	336
Kneser-Ney back-off	3							334	323
Kneser-Ney back-off	4							332	321
Kneser-Ney back-off	5							332	321
class-based back-off	3	150						348	334
class-based back-off	3	200						354	340
class-based back-off	3	500						326	**312**
class-based back-off	3	1000						335	319
class-based back-off	3	2000						343	326
class-based back-off	4	500						327	312
class-based back-off	5	500						327	312

Word2Vec

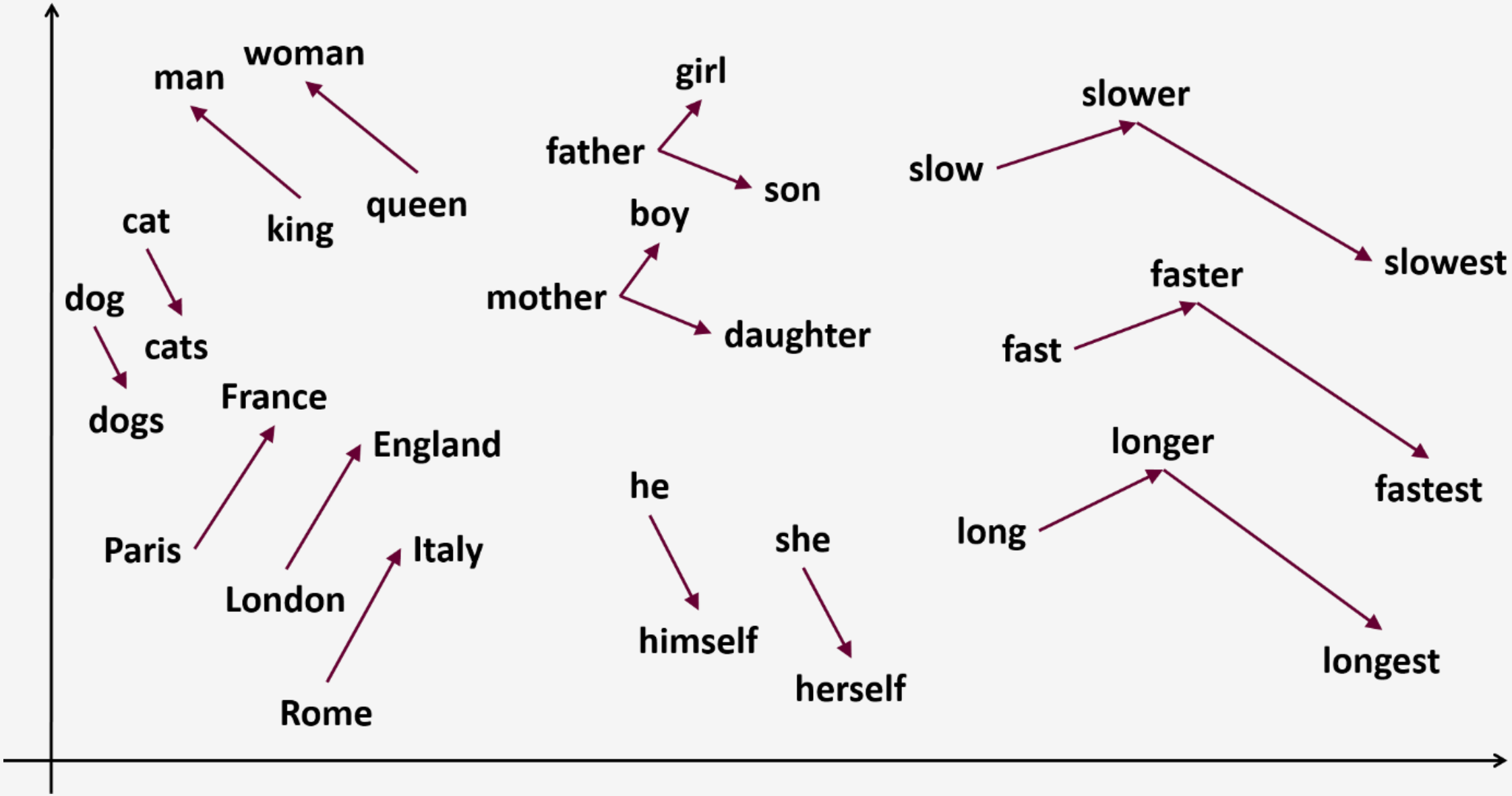
- Mikolov et al (2013)은 Bengio et al (2003)의 연구에서 단어 임베딩에만 초점을 맞췄다.
- 많은 계산을 요하는 인공신경망 대신 logistic classification을 도입했다.
- 주요 알고리즘(with Skip-gram):
  1. Treat the target word and a neighboring context word as positive examples.
  2. Randomly sample other words in the lexicon to get negative samples.
  3. Use logistic regression to train a classifier to distinguish those two cases.
  4. Use the regression weights as the embeddings.



CBOW

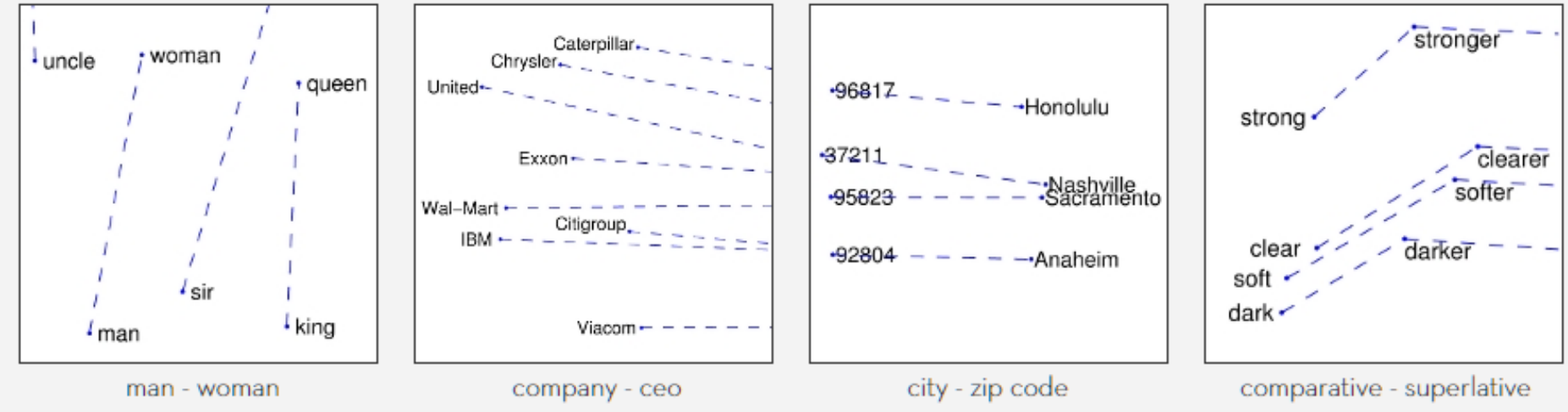


Skip-gram


man
woman
king
queen
cat
cats
dog
dogs
girl
father
son
boy
mother
daughter
slow
slower
slowest
faster
fast
fastest
France
England
Paris
London
Italy
Rome
he
himself
she
herself
long
longer
longest

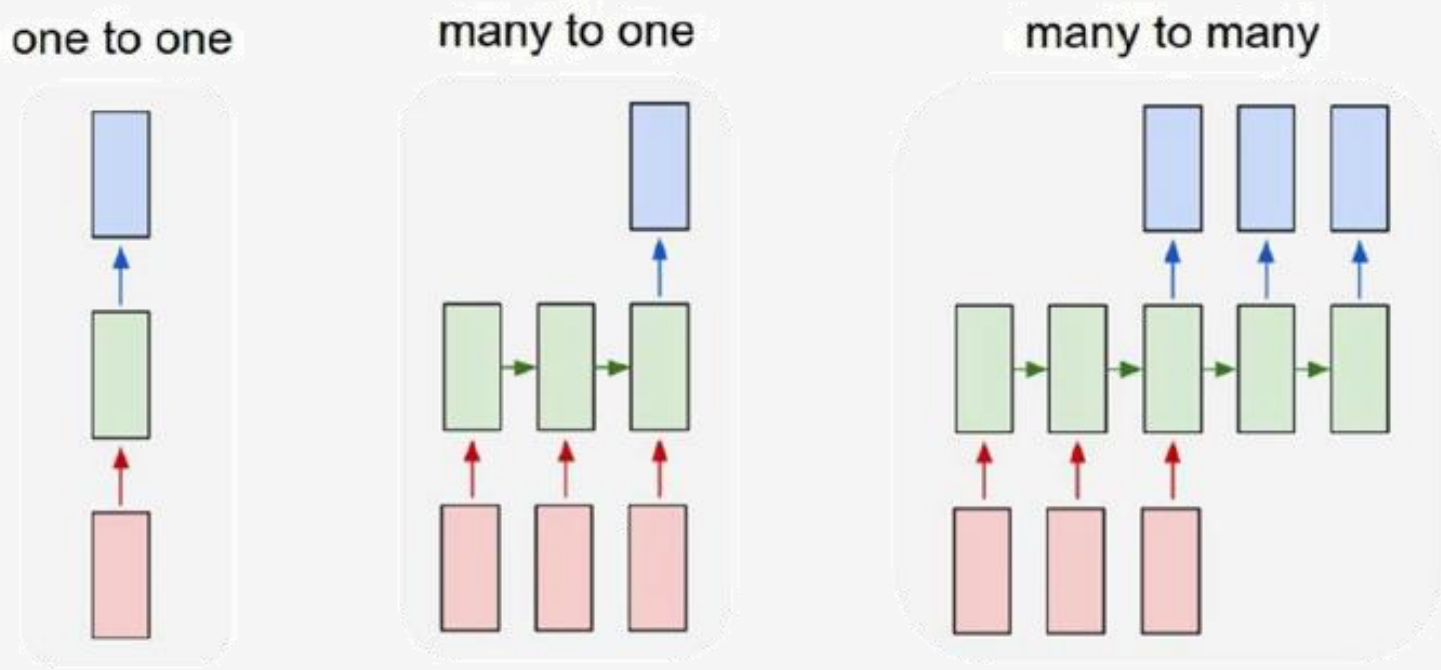
GLoVe

- Pennington et al (2014)는 단어들의 공통 출현 빈도를 이용한 방법을 사용했다.
- 두 단어의 벡터 곱을 두 단어간의 유사도로 보고 이 벡터 곱이 두 단어의 공통출현 확률에 근사하도록 예측모형을 학습시키는 전략을 사용했다.
- GLoVe는 두 단어의 공통출현 관계를 Word2Vec 보다는 더 잘 예측하지만 컴퓨터 계산에서의 부담이 더 크다는 단점이 있다.



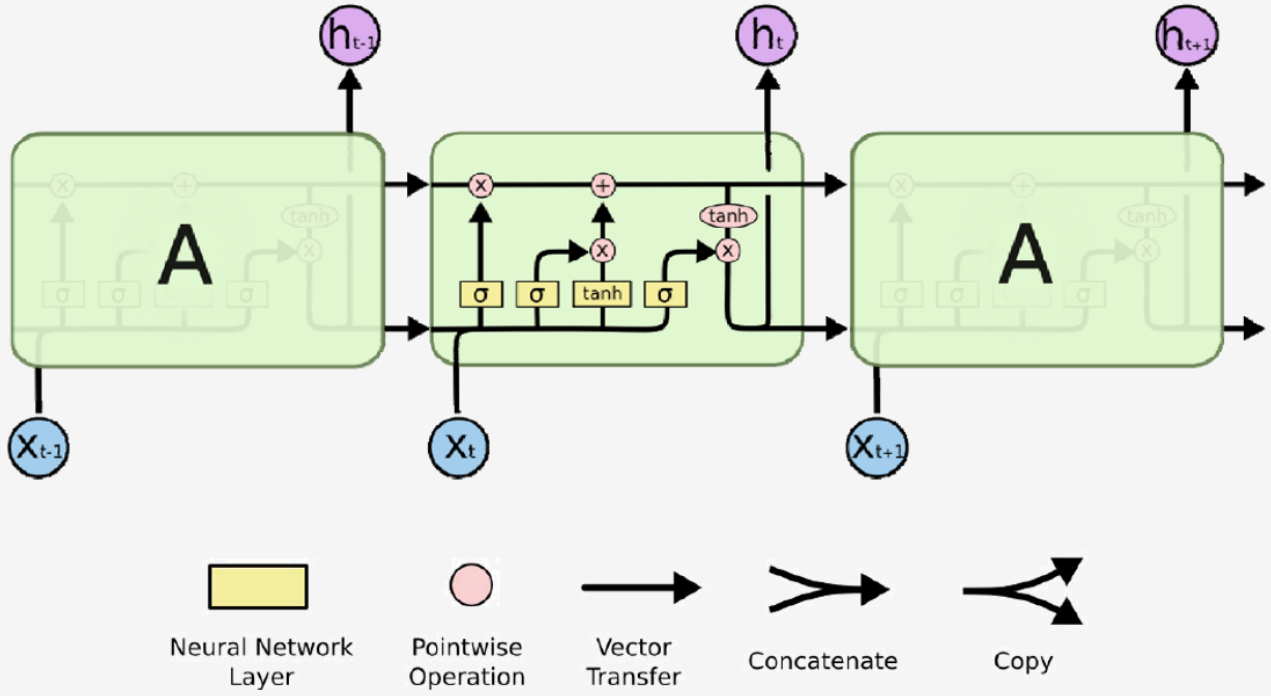
Recurrent Neural Network

- Recurrent Neural Network (RNN)가 자연어 처리 과제에 많이 활용된다.
- RNN은 다양한 유형의 데이터에 대해 다양한 방식의 출력을 줄 수 있다.



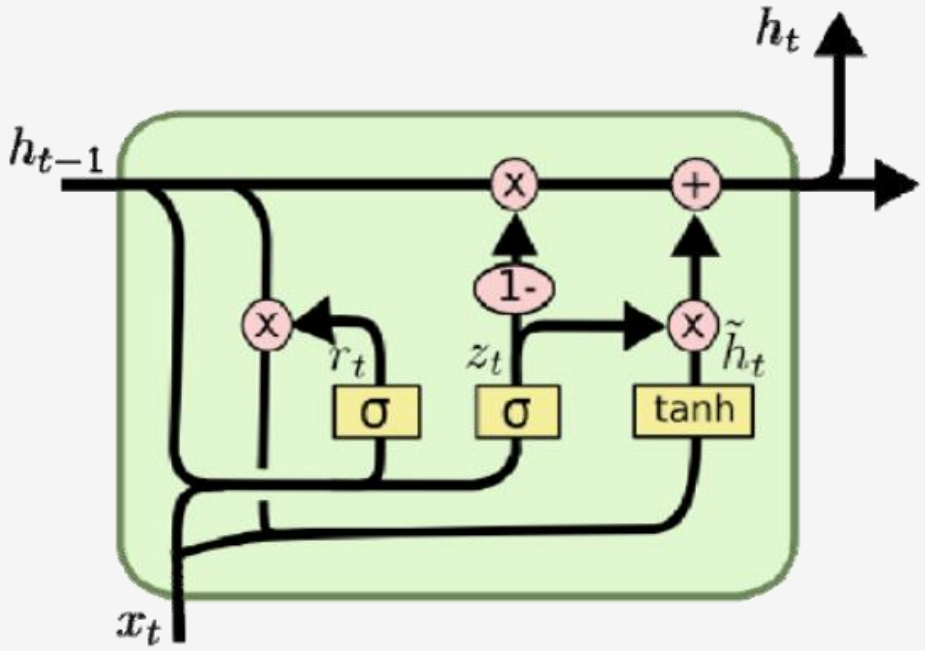
Long-Short Term Memory (LSTM)

- 단순한 RNN은 장기적 의존관계를 제대로 예측하지 못한다는 문제점이 있었다.
- 하지만 자연어에서는 “The man who built the great machine”와 같은 긴 명사절이 있다고 해도 그 다음에 오는 동사는 ‘man’과 의존관계를 맺는 현상이 흔하다.
- Hochreiter & Schmidhuber (1997)의 LSTM 모형은 이런 장거리 의존관계 학습에 적합했다.



Gated Recurrent Unit (GRU)

- GRU도 LSTM과 유사한 아이디어를 통해 장거리 의존관계를 학습한다.
- 구조가 LSTM보다 단순해 계산시간에서 이점이 있다.



Encoder-Decoder Model

- Sutsever et al (2014)에서 Seq2Seq 모형이 제시됨
- 이 계열의 모형은 주로 LSTM 모형을 이용해 기계번역의 성능을 크게 향상시켜 주목을 얻음

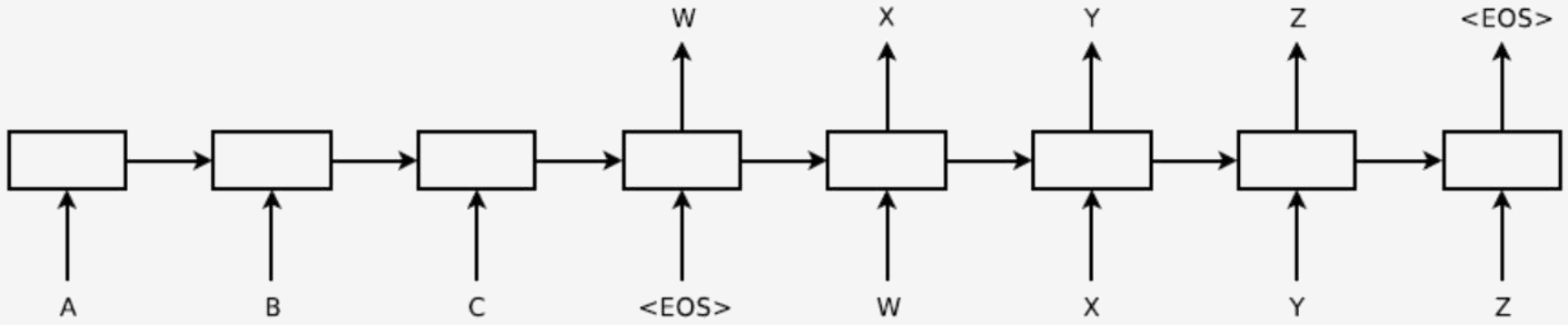
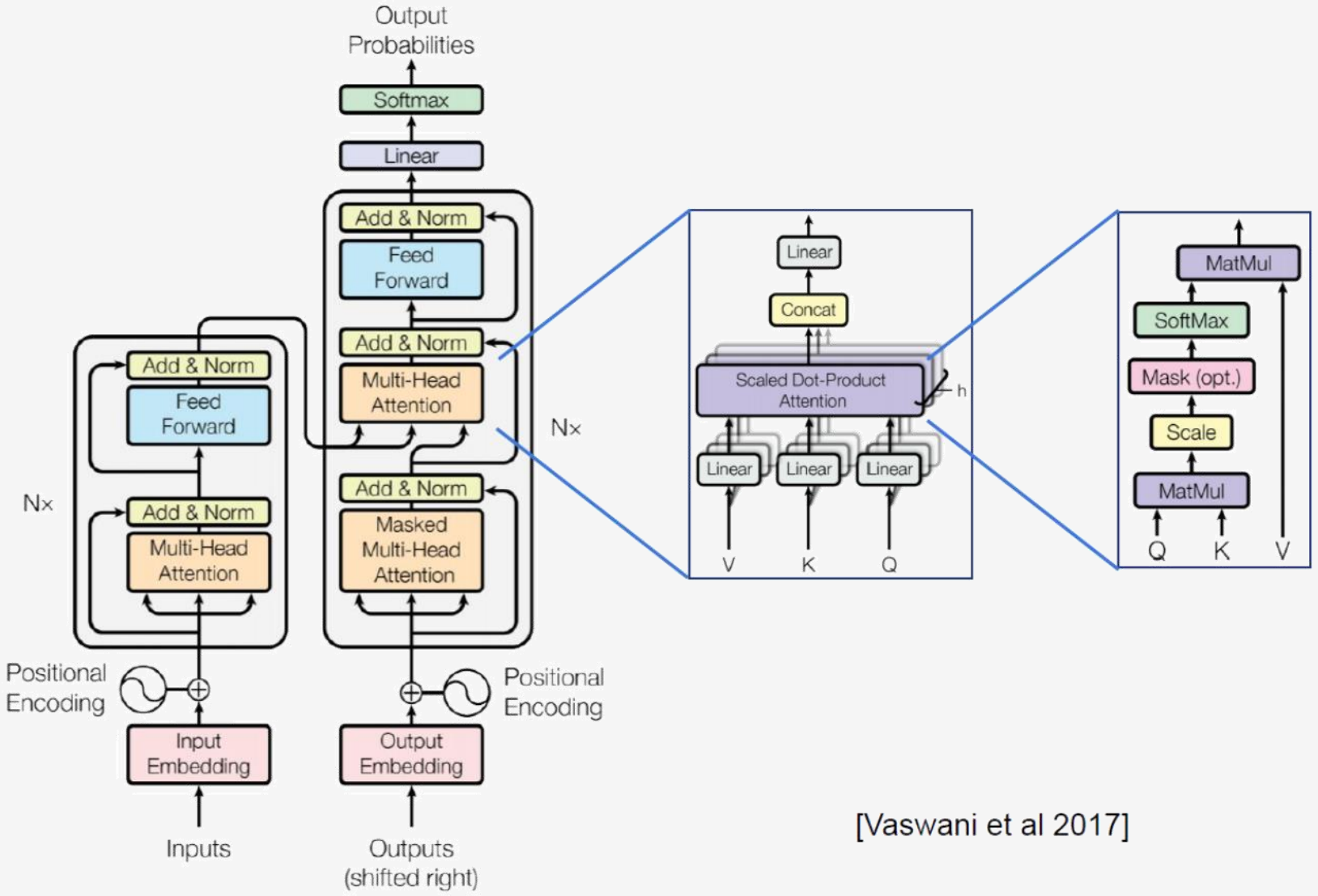


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

RNN 모형의 발전

- Generating Text with Recurrent Neural Networks (Sutskever et al, 2011)
- Skip-Thought Vectors (Kiros et al, 2015)
- Neural Machine Translation by Jointly Learning to Align and Translate (Bahdanau et al, 2015) → Attention 메커니즘 등장
- Deep contextualized word representations (Peters et al, 2018) → ELMO 등장

Transformer: "Attention is All you Need"



[Vaswani et al 2017]

Transformer-based approach

- Transformer 기반 언어모형의 등장
  - BERT (Devlin et al, 2018)
  - GPT-2 (Radford et al, 2019)
- 이 연구들을 통해 deep contextualized pretrained embeddings 를 통해 다양한 자연어 처리 과제들을 처리하는 방법이 널리 퍼졌다.
  - Transfer learning
  - Down-stream tasks
- 위 접근법은 또한 아래와 같은 방법들을 동반하게 된다.
  - Byte-pair encoding (형태소 분석 모듈 필요성 제거)
  - Multi-task learning (순차적 NLP 파이프라인 필요성 제거)

ELMo

ULMFiT

Transformer

BERT

GPT

Bidirectional LM

Larger model
More data

GPT-2

Defense

Grover

ELECTRA

Use GAN for masking

Cross-lingual

XLM
UDify

Multi-task
+ Generation

MT-DNN

Knowledge distillation

$MT\text{-}DNN_{KD}$

MASS
UniLM

Span prediction
Remove NSP

SpanBERT

Longer time
Remove NSP
More data

RoBERTa

Permutation LM
Transformer-XL
More data

XLNet

+Knowledge Graph

ERNIE
(Tsinghua)

Neural entity linker

KnowBert

Cross-modal

VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMBERT
VL-BERT
UNITER

Whole Word Masking

ERNIE (Baidu)
BERT-wwm

By Xiaozhi Wang & Zhengyan Zhang @THUNLP

NLP의 미래?

"You know nothing, Jon Snow"

Ygritte, Game of Thrones

현재 딥러닝 기반 NLP에 대한 불만들

- 설명력(explainability) 부재
  - "뉴럴넷에 X를 넣었는데 이런 Y가 나왔다. 왜 그런가?"
- 부족한 추론(reasoning) 능력
  - 인공신경망을 통한 다단계 추론은 여전히 심볼릭 기반 시스템에 비해 부족하다.
- 높은 시스템 학습 비용
  - RTX 2080 Ti 현재가: ~300만원
  - 저렴하지 않은 클라우드 컴퓨팅 가격

p3.2xlarge	8	31	61GiB	EBS 전용	시간당 4.234 USD
p3.8xlarge	32	97	244GiB	EBS 전용	시간당 16.936 USD
p3.16xlarge	64	201	488GiB	EBS 전용	시간당 33.872 USD

Rebooting AI (Marcus & Davis, 2019)

Tesla driver dies in first fatal crash while using autopilot mode

The autopilot sensors on the Model S failed to distinguish a white tractor-trailer crossing the highway against a bright sky

▲ Joshua Brown, the first person to die in a self-driving car accident. Photograph: Facebook

Against a bright spring sky, the car's sensors system failed to distinguish a large white 18-wheel truck and trailer crossing the highway, Tesla said. The car attempted to drive full speed under the trailer, "with the bottom of the trailer impacting the windshield of the Model S", Tesla said in a blogpost.

A police report in the Levy County Journal said the top of the vehicle "was torn off by the force of the collision". The truck driver, Frank Baressi, 62, of Tampa, Florida, was uninjured, the Journal reported.

America's National Highway Traffic Safety Administration (NHTSA) has opened an inquiry into the accident.

Brown owned a technology company called Nexu Innovations and was a Tesla enthusiast who posted videos of his car on autopilot on YouTube. One video showed his car avoiding a collision on a highway, racking up 1m views after it was tweeted by Tesla CEO Elon Musk.

가능한 대안?

- Deep Neural Connectome
  - Deep neuroethology of a virtual rodent (ICLR, 2020)
  - OpenAI Microscope project
- Graph Neural Networks
  - The Logical Expressiveness of Graph Neural Networks (ICLR, 2020)
- Neuro-Symbolic approach
  - Neural Symbolic Reader: Scalable Integration of Distributed and Symbolic Representations for Reading Comprehension (ICLR, 2020)
- Distillation, Pruning and Quantization
  - DistilBERT (ArXiv, 2019)

References

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In ICLR 2015 : International Conference on Learning Representations 2015.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of Machine Learning Research, 3(Feb), 1137–1155.

Chomsky, Noam (1956), "Three models for the description of language", IEEE Transactions on Information Theory, 2 (3): 113–124.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., & Fidler, S. (2015). Skip-thought vectors. In NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (pp. 3294–3302).

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In ICLR (Workshop Poster).

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.

Schütze, H. (1992). Word Space. In Advances in Neural Information Processing Systems 5 (pp. 895–902).

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating Text with Recurrent Neural Networks. In Proceedings of the 28th International Conference on Machine Learning (pp. 1017–1024).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In Advances in Neural Information Processing Systems 27 (pp. 3104–3112).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is All you Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 5998–6008).

THANKS