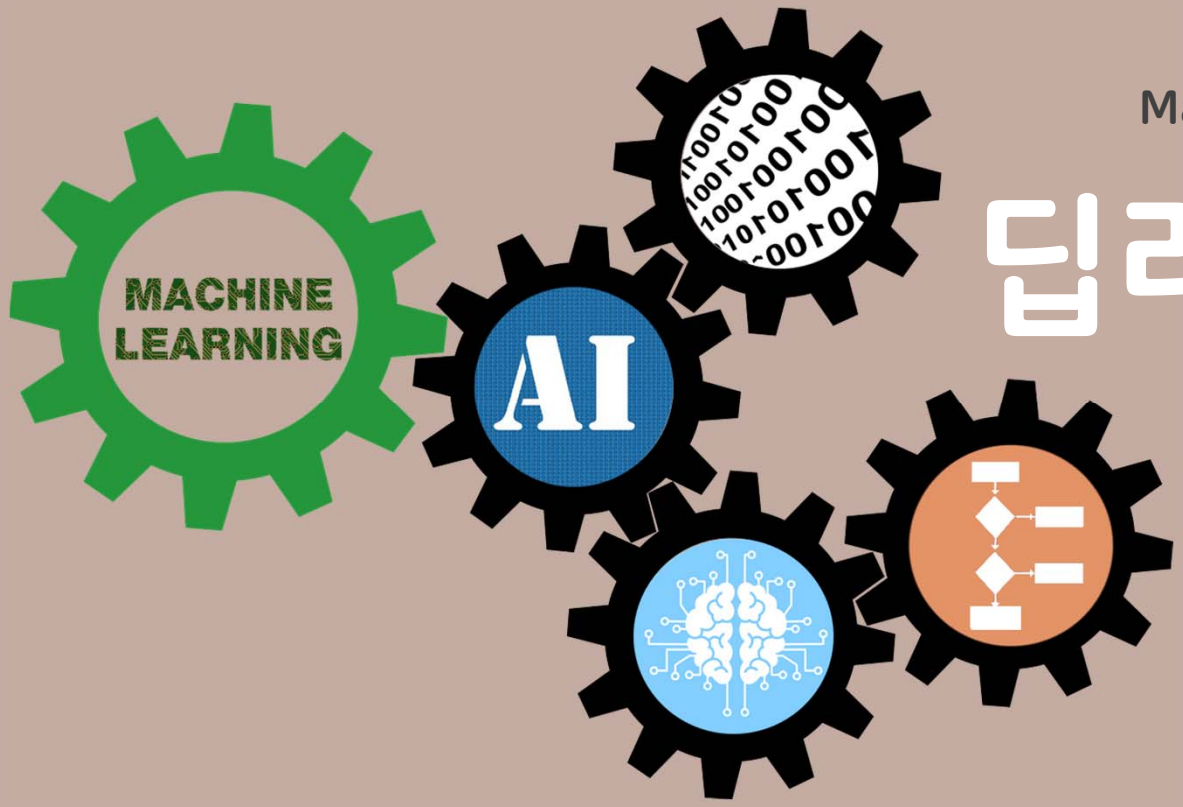


Machine Learning 14강

딥러닝 응용 (2)

컴퓨터과학과 이관용 교수

학습목차

1

자연어처리 응용

자연어처리

- NLP, Natural Language Processing
  - 자연어 → 한국어, 영어, 스페인어, 중국어, 일본어 등
    - ✓ 인간언어 → 인공언어에 대응되는 개념
    - ✓ 정보 전달의 수단으로 인간 고유의 능력
  - 인공언어 artificial language → 예: 프로그래밍 언어(C, Python, Java 등)
    - ✓ 특정 목적을 위해 인위적으로 만든 언어
    - ✓ 자연어보다 엄격한 문법을 가짐
  - 자연어처리
    - ✓ 컴퓨터로 자연어를 이해understanding하고, 번역interpret하고, 조작manipulate하기 위한 인공지능의 한 분야

자연어처리의 구성 요소

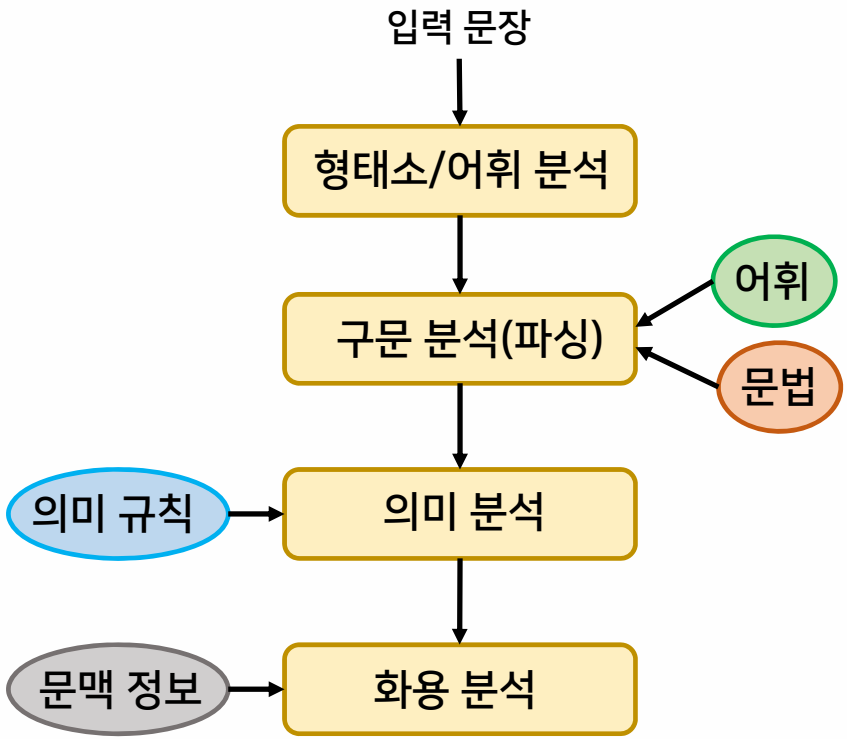
형태소/어휘 분석
Morphological and Lexical Analysis

구문 분석
Syntactic Analysis

의미 분석
Semantic Analysis

담론 통합
Discourse Integration

화용(어용) 분석
Pragmatic Analysis



https://www.guru99.com/nlp-tutorial.html#1

NLP의 구성 요소 https://www.guru99.com/nlp-tutorial.html#1

형태소 분석

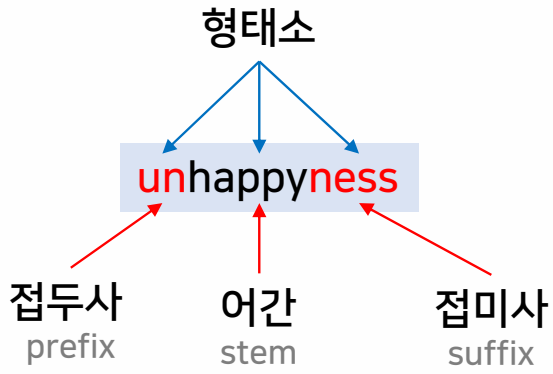
- 문장을 이루는 최소 의미 단위인 형태소morpheme로 분리

감기는

→ 감기(명사) + 는(조사)

→ 감(동사어간) + 기(명사화어미)+는(조사)

→ 감(동사어간) + 기는(어미)

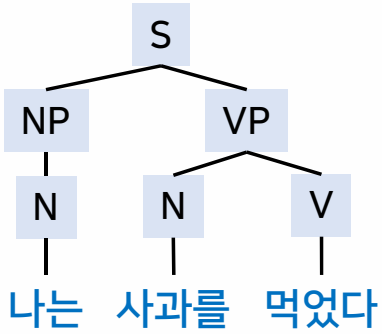


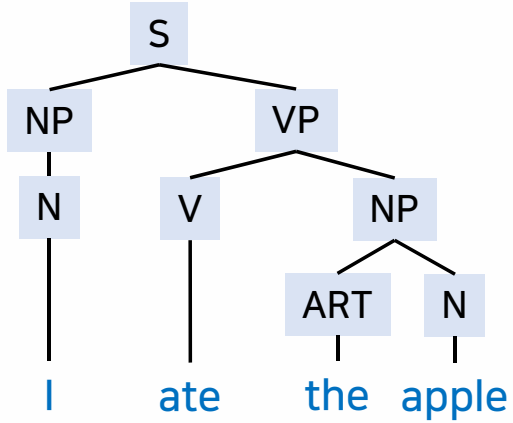
- 형태소 분석의 어려움 → 중의성 ambiguity, 접두사/접미사 처리
- 언어에 따라 난이도가 달라짐

NLP의 구성 요소 https://www.guru99.com/nlp-tutorial.html#1

구문 분석 파싱

- 주어진 문장의 구조를 문법에 맞춰 분석
- 문법 → 문장의 구조적 성질을 규칙으로 표현한 것





- 어려움 → 구조적 중의성 structural ambiguity

Time flies like light.

A man see a woman with a telescope.

NLP의 구성 요소 https://www.guru99.com/nlp-tutorial.html#1

의미 분석

- 구문 분석 결과에 의미를 가하여 문장이 가진 의미를 분석
- 형태소가 가진 의미를 표현하는 지식표현 기법이 필요
- 문법은 맞으나 의미적으로 틀린 문장을 검사

  원숭이가 사과를 먹는다.

  기차가 구름을 먹는다.

- 어려움 → 의미적 중의성

  말이 많다. → many horses

  → chatty

NLP의 구성 요소 https://www.guru99.com/nlp-tutorial.html#1

화용 분석

- 문장이 실제로 사용될 때 연관관계를 분석
- 담화 분석 discourse analysis
  - ✓ 상호참조 coreference → 대명사가 지시하는 대상 확인

    John's boss said he was getting better.

  - ✓ 화행 speech act 분석 → 발화의 의도 분석(정보요구, 정보제공, 거절 등)

    Can you give me a salt?

- 실세계의 지식 표현이 필요

자연어처리의 어려움

- 문법과 규칙에 기반 → 많은 예외사항이 존재
- 모호성(중의성) 존재
  - 문맥 정보와 엄청난 양의 지식이 필요
- 높은 차원
  - 많은 개수의 단어가 사용 → 효율적인 표현 방법이 필요
- 순차적인 입력과 출력
  - 시퀀스가 중요 → 시퀀스 처리 능력이 필요

⇒ 머신러닝/딥러닝이 자연어처리의 좋은 도구가 될 수 있음

자연어처리 응용

- 음성 신호처리 speech signal processing
  - 음성인식 speech recognition, 화자인식 speaker recognition, 음성합성
- 대화 수행 dialogue action
  - 정보검색 information retrieval, 질의응답 question-answering,
    목적 지향 대화 task-oriented dialogue
- 텍스트 분석 text analysis
  - 텍스트 분류 text classification
    - ✓ 스팸 필터링, 감성 인식 sentiment classification, 주제 분류 text categorization
  - 기계번역 machine translation , 텍스트 요약 text summarization

음성 신호처리

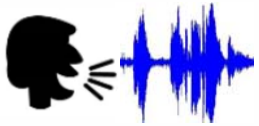
- 사람의 목소리로 발화된 음성 신호에 포함된 언어 정보를 처리

음성인식

음성 신호 → ASR 시스템 → 텍스트

speech-to-text 변환

화자인식



→ 화자인식 →  (발화자의 신원)

음성합성

→ 음성합성 → 

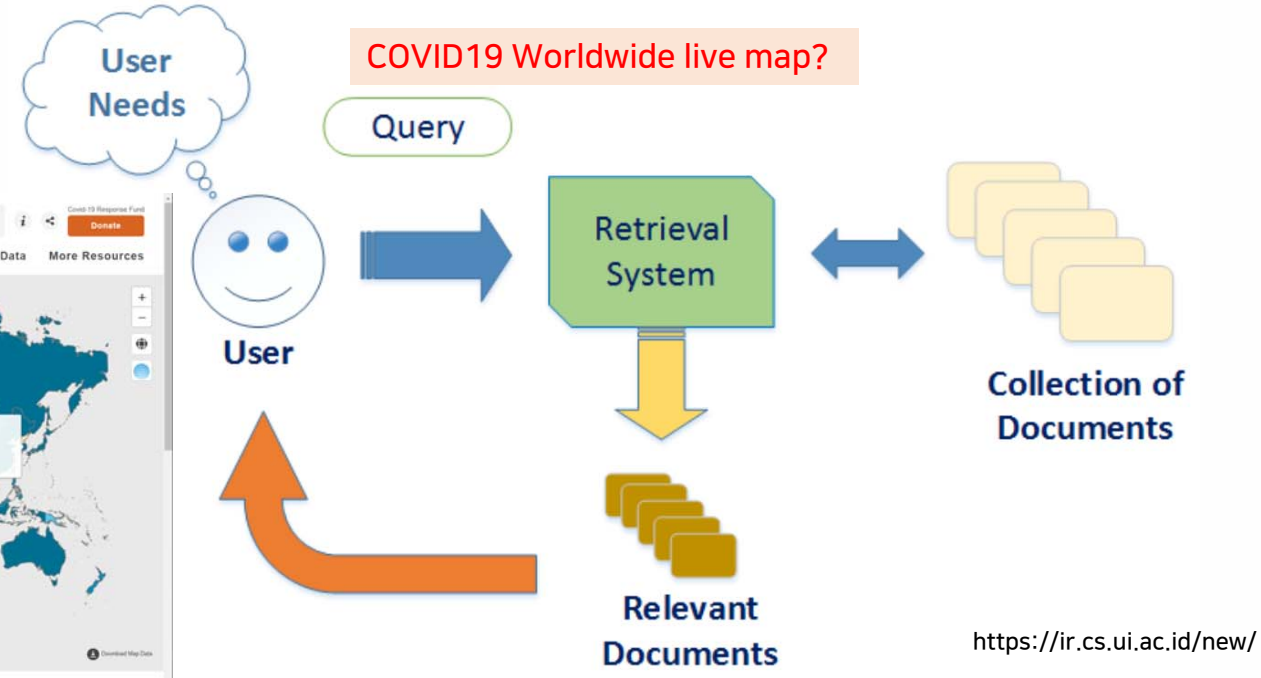
TTS (Text To Speech)

이미지 출처 https://www.slideshare.net/YunNungVivianChen/endtoend-taskcompletion-neural-dialogue-systems

대화 수행

- 상대방과 문장을 주고받는 형식으로 진행되는 작업
  - AI 스피커, 챗봇 등의 핵심 기술

정보검색



https://ir.cs.ui.ac.id/new/

대화수행

질의응답 시스템

Read-and-Search Process for End-to-End Open-Domain QA

Q. What is the largest island in the Philippines?

WIKIPEDIA

IR Model

Paragraphs:

1) Mindanao is the second largest and easternmost island in the Philippines.
2) As an island, Luzon is the Philippine's largest at 104,688 square kilometers, and is also the world's 17th largest island.
3) Manila, located on east central Luzon Island, is the national capital and largest city.
4) ……

Ranker

As an island, Luzon is the Philippine's largest at 104,688 square kilometers, and is also the world's 17th largest island.

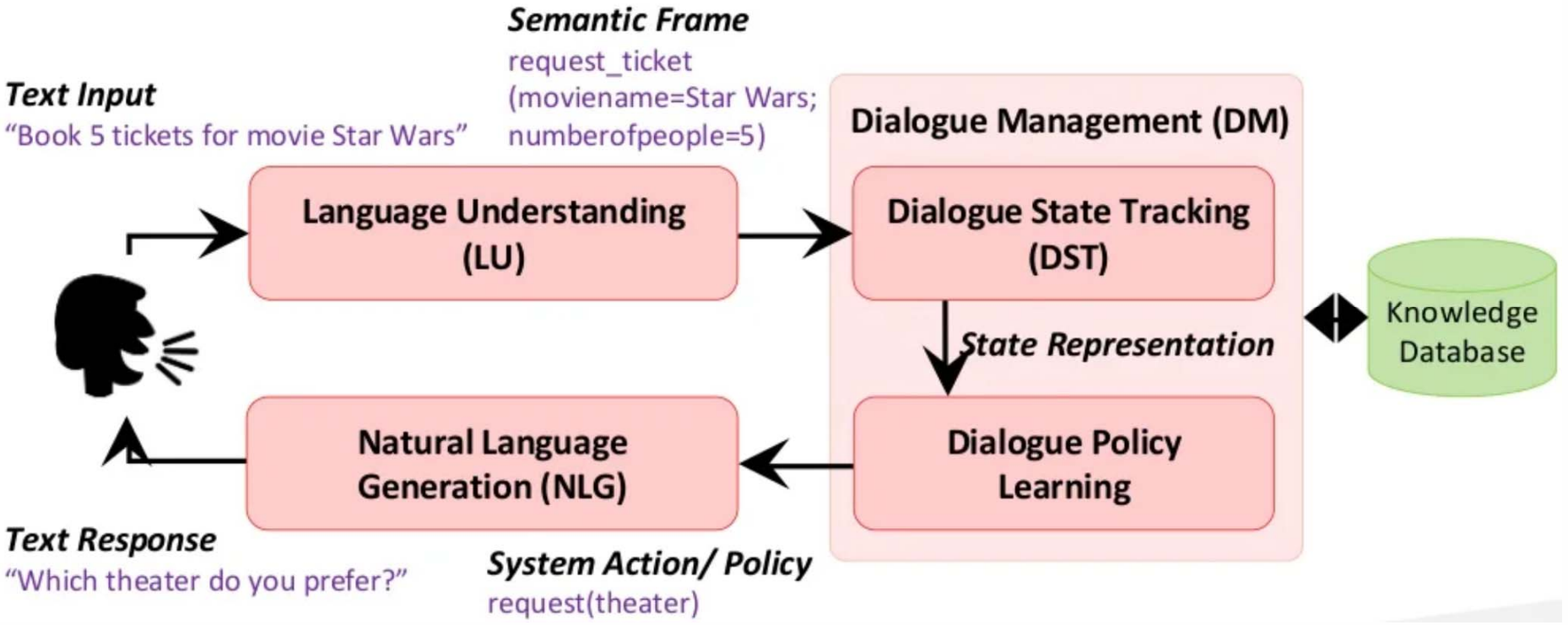
Reader

Luzon

https://www.ibm.com/blogs/research/2018/02/open-domain-qa/

대화수행

목적 지향 대화 시스템



https://www.slideshare.net/YunNungVivianChen/endtoend-taskcompletion-neural-dialogue-systems

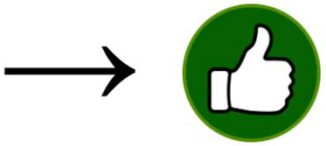
텍스트 분석

- 일련의 텍스트 정보를 입력으로 받아 그 의미적 내용과 문맥 등을 분석하여 원하는 결과를 도출
  - 텍스트 분류
    - ✓ 주어진 일련의 텍스트를 미리 정해진 몇 개의 클래스로 나누는 것
    - ✓ 감성인식, 주제 분류, 스팸 필터링 등
  - 기계번역
    - ✓ 문장을 입력받아 같은 의미를 가진 다른 언어의 문장을 생성
  - 텍스트 요약
    - ✓ 긴 분량의 텍스트를 입력받아 짧게 요약된 문장으로 출력

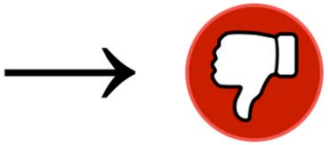
텍스트 분석: 텍스트 분류의 응용 예

감성인식

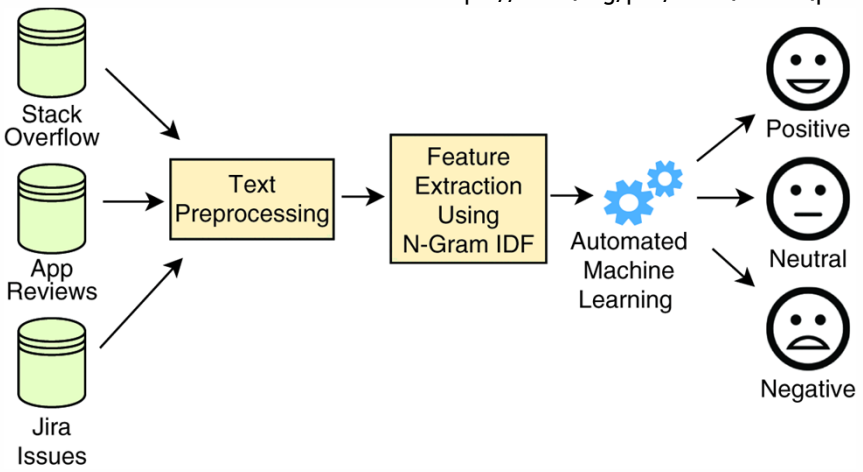
"I love this movie.
I've seen it many times
and it's still awesome."



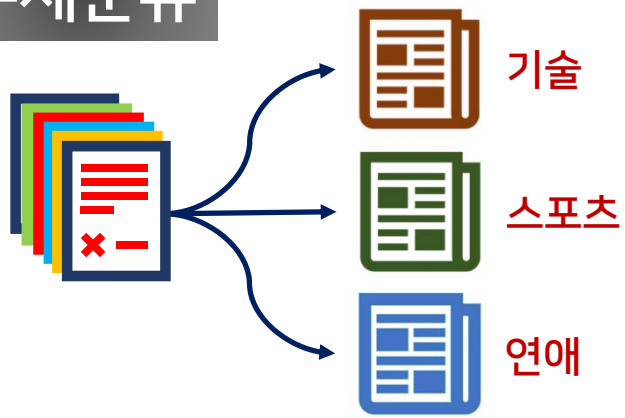
"This movie is bad.
I don't like it it all.
It's terrible."



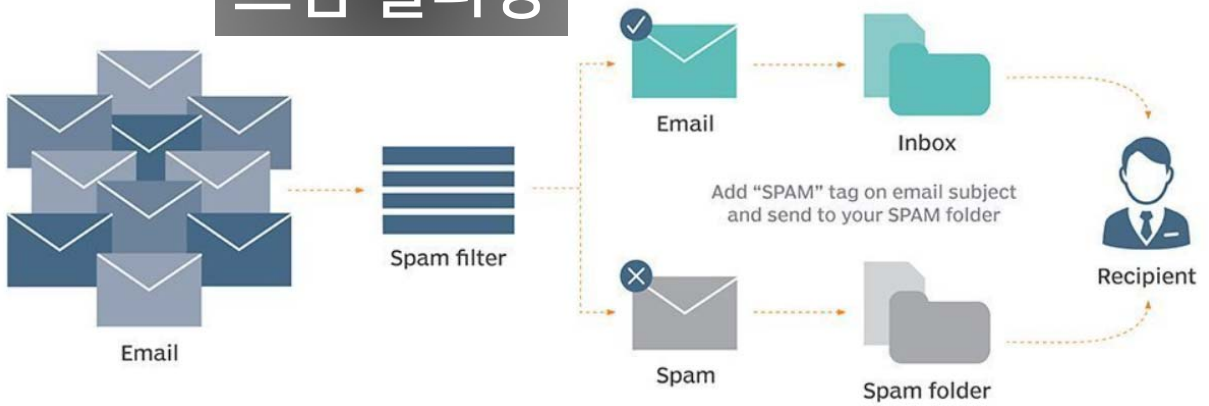
https://arxiv.org/pdf/1904.12162.pdf



주제분류

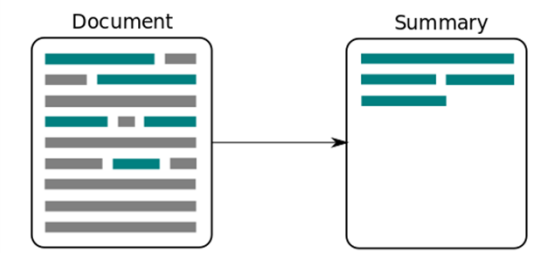


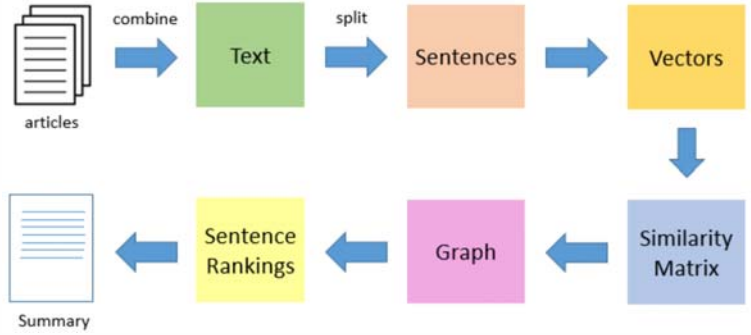
스팸 필터링



https://www.techtarget.com/searchsecurity/definition/spam-filter

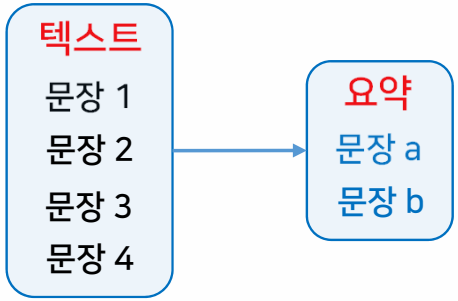
텍스트 분석: 텍스트 요약

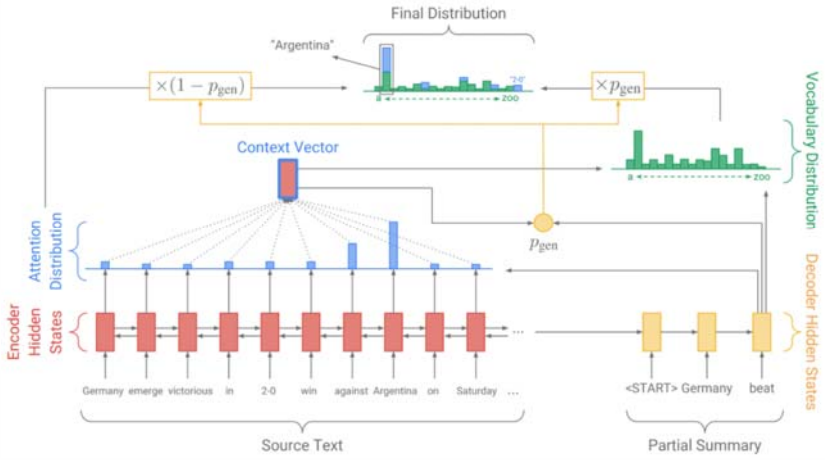




추출 extractive 요약 → 가장 적절한 문장을 선택하여 조합

추상적 abstractive 요약 → 새로운 문장을 사용하여 요약

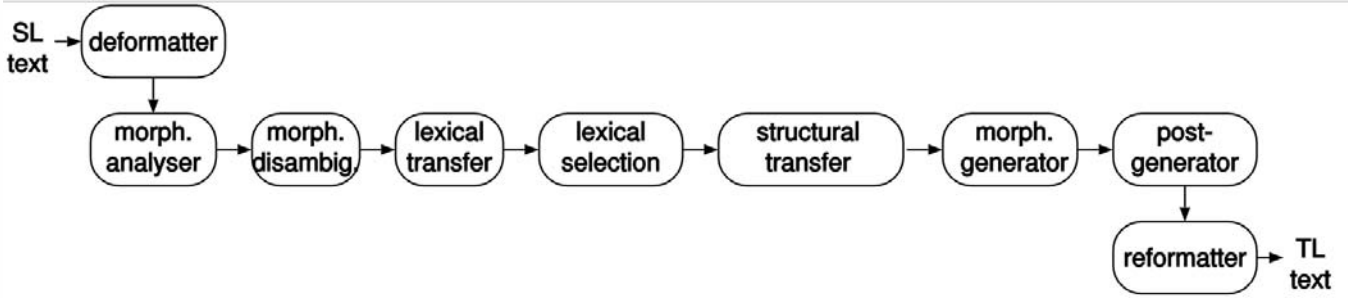




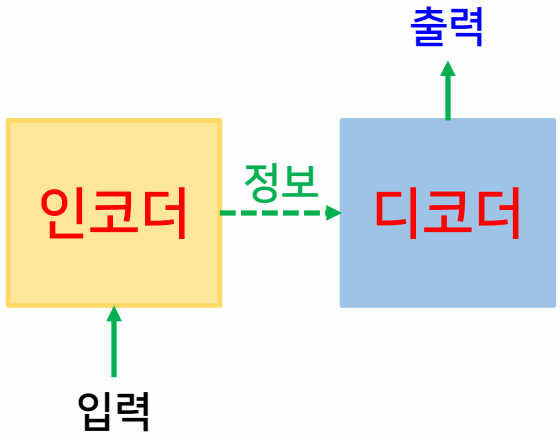
https://towardsdatascience.com/data-scientists-guide-to-summarization-fc0db952e363
https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

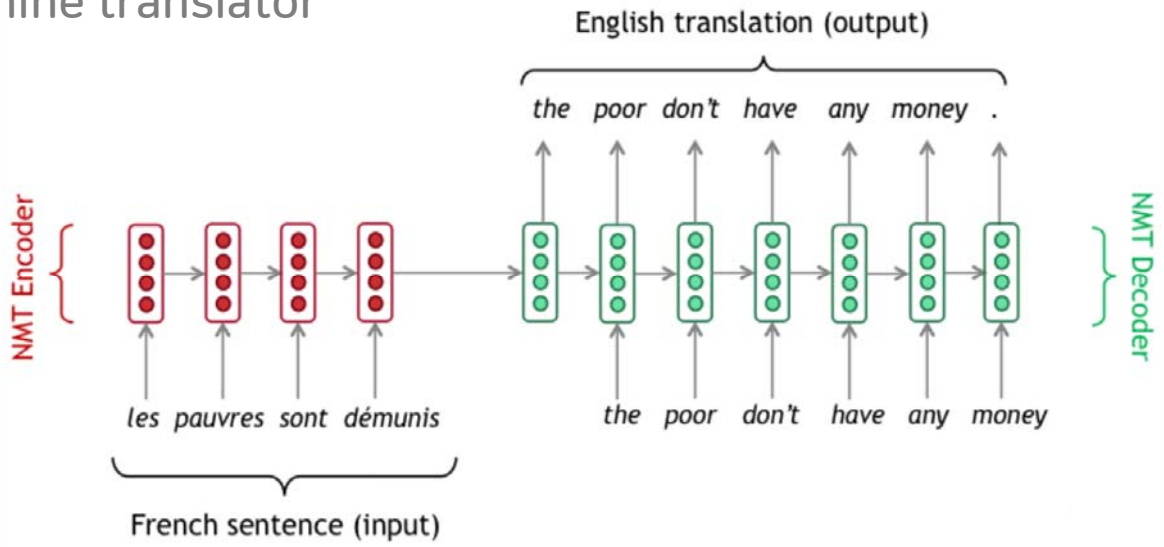
텍스트 분석: 기계번역

- 고전적인 접근법



- 딥러닝 접근법 "neural machine translator"

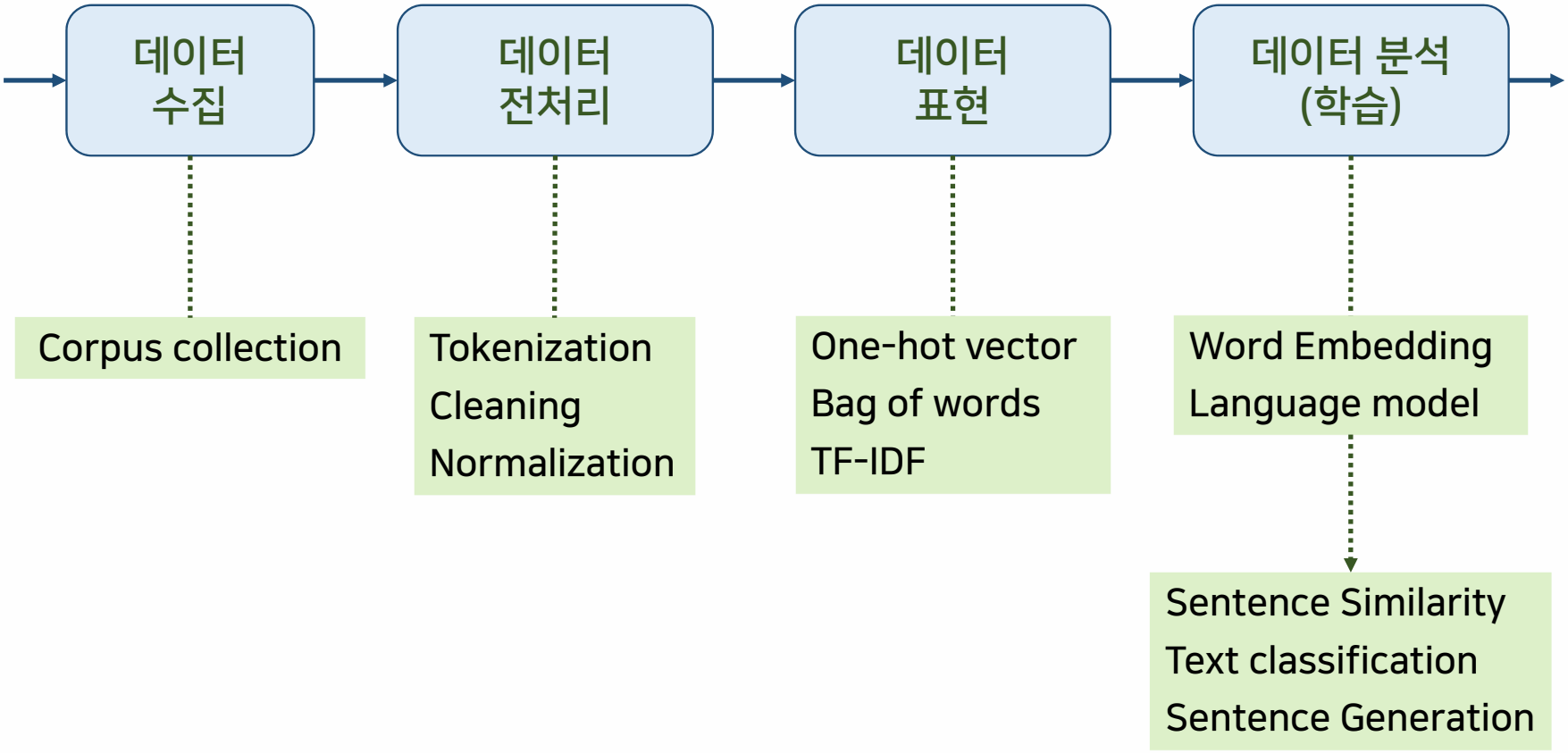




https://medium.com/@vivekvscool/translation-or-answer-tool-seq2seq-with-teacher-forcing-and-attention-mechanism-7cfd9cb03b3a

2 자연어처리를 위한 머신러닝 기법

NLP를 위한 ML 시스템 개발 단계



NLP를 위한 데이터 수집

- 텍스트 말뭉치 text corpus
  - 크고 구조화된 텍스트 데이터 집합
  - 주요 말뭉치
    - ✓ 구글 n-gram 말뭉치
      http://commondatastorage.googleapis.com/books/syntactic-ngrams/index.html
    - ✓ COCA Corpus of Contemporary American English https://www.english-corpora.org/coca/
      → 4억2,500만개 단어, 1990~2011, 무료 온라인 검색 서비스 제공
    - ✓ 국립국어원 '모두의 말뭉치' https://corpus.korean.go.kr/main.do
  - WordNet https://wordnet.princeton.edu/
    - ✓ a lexical database for English

텍스트 전처리

토큰화 tokenization

- 말뭉치를 의미 있는 기본 단위("token")으로 나누는 작업
  - ✓ 토큰의 기준 → 단어, 문장, 구 phrase, 형태소

Time is an illusion. —토큰화→ "Time", "is", "an", "illusion"

- 고려사항
  - ✓ 구두점, 특수문자의 처리 → 예: $50.50, 22/10/04, AT&T, Ph.D 등
  - ✓ 줄임말, 단어 내 띄어쓰기
  - ✓ 한국어 → 조사, 어간과 어미 분리

철수는 책을 읽었다. → 자립형태소(철수, 책), 의존형태소(는, 을, 읽-, 었-, -다)

텍스트 전처리

정제 cleaning, 정규화 normalization

- 정제 → 말뭉치로부터 데이터 분석에 방해되는 노이즈 데이터 제거
  - ✓ 불필요한 단어 제거 → 등장 빈도가 적은 단어, 길이가 짧은 단어(예: it, at, to, on, in, by, …)
  - ✓ 정규표현식을 사용하여 특정 표현 제거 → 예: 해시태그, 기사의 날짜 등
- 정규화 → 표현 방법이 다른 단어들을 하나의 단어로 통합시키는 것
  - ✓ 표기가 다른 단어들의 통합 → 예: US, USA
  - ✓ 대소문자 통합

토큰의 품사 태깅 작업

열심히 공부한 당신, 연휴에는 여행을 가봐요

↓

[(‘열심히’,’부사’), (‘공부’,’명사’), (‘한’,’조사’), (‘당신’,’명사’), (‘,’,’구두점’),
(‘연휴’,’명사’), (‘에는’,’조사’), (‘여행’,’명사’), (‘을’,’조사’), (‘가봐요’,’동사’)]

데이터 표현

원핫인코딩 one-hot encoding

- 말뭉치로부터 획득한 단어집합 vocabulary의 각 단어를 고유 정수로 매핑한 후 원핫벡터로 표현
- m개의 단어가 있는 경우 → m차원 원핫벡터

Corpus	
문서1	먹고 싶은 수박
문서2	먹고 싶은 토마토
문서3	크고 무거운 수박 수박
문서4	과일 가격이 비싸다

One-hot Encoding for Words	
먹고	[1,0,0,0,0,0,0,0,0]
싶은	[0,1,0,0,0,0,0,0,0]
수박	[0,0,1,0,0,0,0,0,0]
토마토	[0,0,0,1,0,0,0,0,0]
크고	[0,0,0,0,1,0,0,0,0]
무거운	[0,0,0,0,0,1,0,0,0]
과일	[0,0,0,0,0,0,1,0,0]
가격	[0,0,0,0,0,0,0,1,0]
비싸다	[0,0,0,0,0,0,0,0,1]

Indexing

단어	먹고	싶은	수박	토마토	크고	무거운	과일	가격	비싸다
index	0	1	2	3	4	5	6	7	8

- 한계점 → 단어수가 많아지면 차원이 높아짐. 단어 간의 유사도 반영 불가

데이터 표현

BoW, Bag of Words

- 단어의 출현 빈도수 frequency를 고려한 텍스트 표현 방법

  (1) 단어집합에 포함된 각 단어에 고유한 정수 인덱스를 부여

  (2) 주어진 입력 텍스트에 대하여 각 단어의 출현 횟수를 계산

  (3) 각 단어의 대응 위치(인덱스)에 출현 회수를 정수값으로 표현

Corpus

문서1	먹고 싶은 수박
문서2	먹고 싶은 토마토
문서3	크고 무거운 수박 수박
문서4	과일 가격이 비싸다

Indexing

단어	먹고	싶은	수박	토마토	크고	무거운	과일	가격	비싸다
index	0	1	2	3	4	5	6	7	8

입력 텍스트

무거운 수박이 크고 수박 가격이 비싸다

인덱스	0	1	2	3	4	5	6	7	8
빈도	0	0	2	0	1	1	0	1	1

BoW → [0, 0, 2, 0, 1, 1, 0, 1, 1]

- 문서에 자주 출현하는 단어가 잘 표현됨. 단어의 발생 위치는 고려되지 않음

데이터 표현

TF-IDF Term Frequency-Inverse Document Frequency

Document-Term Matrix

✓ 다수 문서에 등장하는 각 단어들의 빈도수를 표현한 행렬

문서1	먹고 싶은 수박
문서2	먹고 싶은 토마토
문서3	크고 무거운 수박 수박
문서4	과일 가격이 비싸다

→

	먹고	싶은	수박	토마토	크고	무거운	과일	가격	비싸다
문서1	1	1	1	0	0	0	0	0	0
문서2	1	1	0	1	0	0	0	0	0
문서3	0	0	2	0	1	1	0	0	0
문서4	0	0	0	0	0	0	1	1	1

$TF(d,t)$ → d번째 문서에 t번째 단어가 나타나는 횟수

Document Frequency

✓ 각 단어가 나타나는 문서의 빈도수를 계산

	먹고	싶은	수박	토마토	크고	무거운	과일	가격	비싸다
$DF(t)$	2	2	2	1	1	1	1	1	1

데이터 표현

TF-IDF Term Frequency-Inverse Document Frequency

- 문서 내의 각 단어의 빈도수와 문서의 빈도를 함께 고려한 표현 방식

$$TF-IDF(d,t) = TF(d,t) \times IDF(t) = TF(d,t) \times log\left(\frac{N}{DF(t)}\right)$$

문서 d에서 단어 t가 나타나는 횟수

단어 t가 등장하는 문서의 개수 $DF(t)$에 반비례하는 값

	먹고	싶은	수박	토마토	크고	무거운	과일	가격	비싸다
문서1	0.301	0.301	0.301	0	0	0	0	0	0
문서2	0.301	0.301	0	0.602	0	0	0	0	0
문서3	0	0	0.602	0	0.602	0.602	0	0	0
문서4	0	0	0	0	0	0	0.602	0.602	0.602

- 특정 문서에 국한된 단어는 큰 값, 일반적인 공용 단어는 낮은 값

데이터 분석

워드 임베딩 word embedding

- 단어의 의미를 포함하는 벡터("임베딩 벡터")로 표현하는 방법
  - ✓ 원핫벡터를 저차원 실수 공간의 벡터로 변환

	원핫벡터	임베딩 벡터
차원	고차원(=단어집합의 크기) → 희소 벡터의 형태	저차원 → 밀집 벡터의 형태
표현 방법	수동	학습 데이터로부터 학습
값	0과 1로 구성	실수값

https://wikidocs.net/33520

- 목적 → 유사한 의미의 단어를 가까운 위치에 표현
- 대표적 방법 → Word2Vec (CBoW, Skip-gram)

Word2Vec

- 원핫벡터를 저차원의 벡터로 변환하는 선형변환행렬 W를 학습으로 찾고, 이를 이용하여 입력 단어를 사영함으로써 임베딩 벡터를 구함
  - 말뭉치의 문맥 정보를 활용하여 학습을 수행
  - 은닉층이 1개인 간단한 구조의 신경망 사용
- 학습 방식에 따라 두 가지 모델이 존재
  - CBoW Continuous Bag of Words
    - ✓ 주변 단어(문맥 앞뒤의 n개 단어)들을 입력으로 받아 중심 단어를 예측
  - Skip-gram
    - ✓ 중심 단어를 입력으로 받아 주변 단어들을 예측

Word2Vec

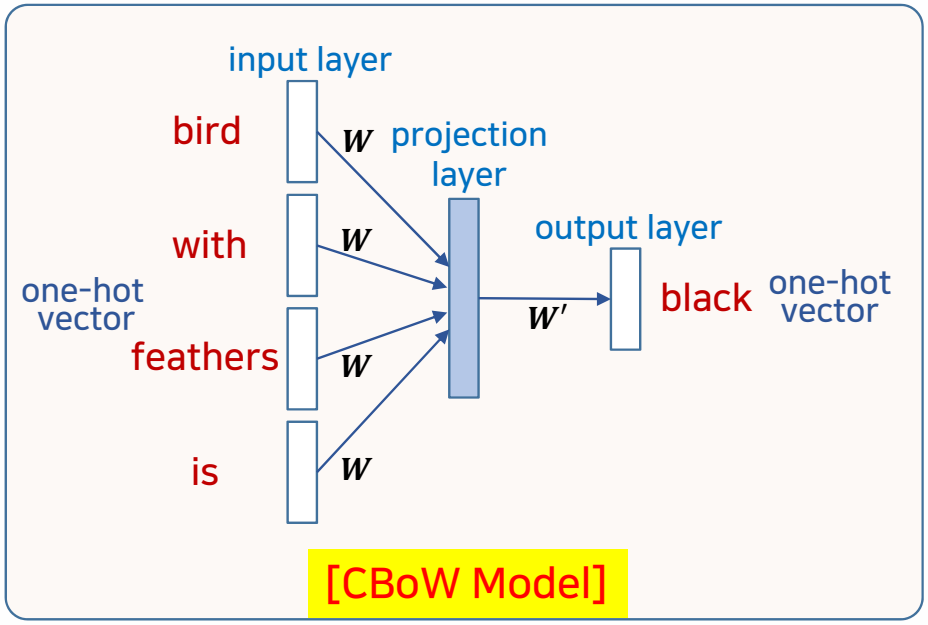
[Corpus]

sliding window ($n = 2$)

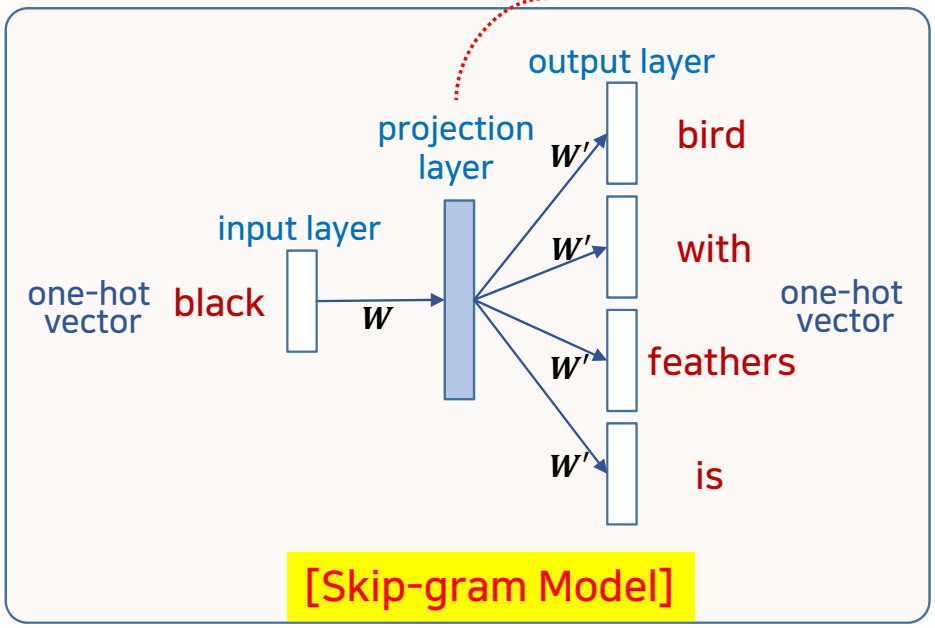
The bird with black feathers is on the tree.

문맥 단어 | 중심 단어 center word | 문맥 단어 context word

일반적인 은닉층과 달리 활성화 함수가 없고,
룩업 테이블이라는 연산을 수행하는 층

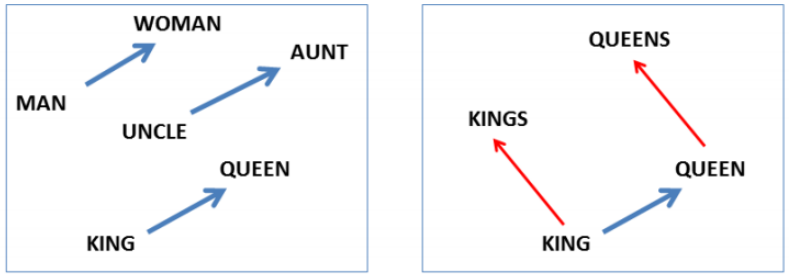


[CBoW Model]

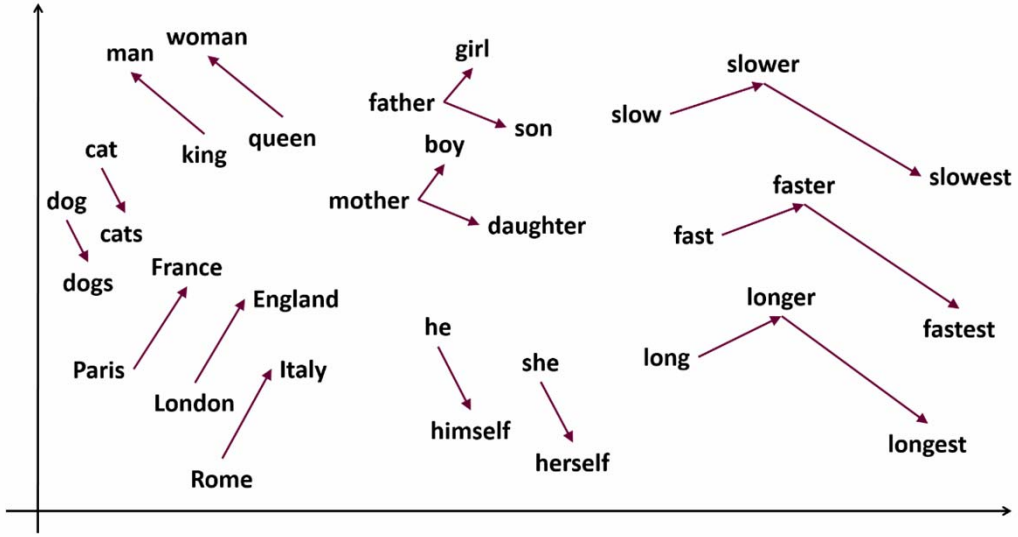


[Skip-gram Model]

Word2Vec의 수행 결과의 예

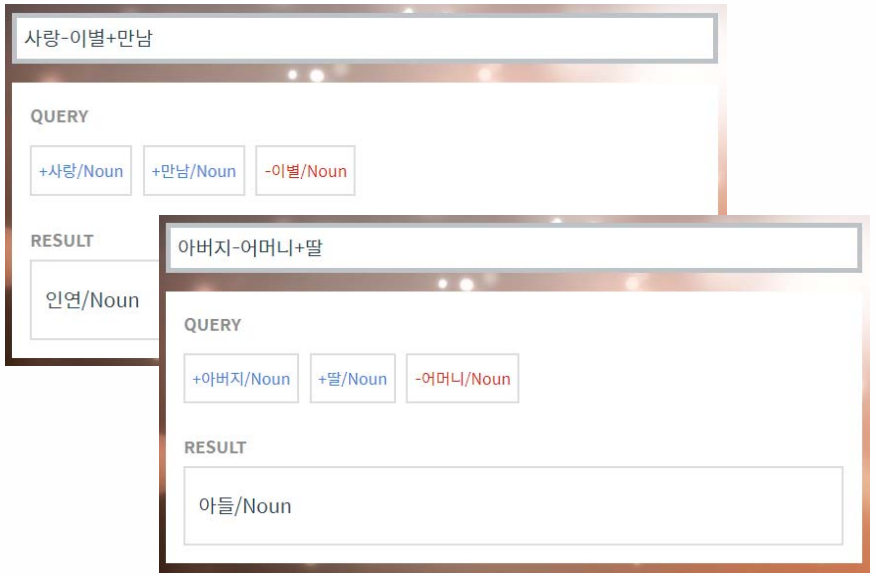


(Mikolov et al., NAACL HLT, 2013)



https://medium.com/analytics-vidhya/implementing-word2vec-in-tensorflow-44f93cf2665f

Korean Word2Vec http://w.elnn.kr/search/



3 언어 모델을 위한 딥러닝

언어 모델

- 단어 시퀀스를 입력으로 받아 확률값을 출력하는 일종의 함수

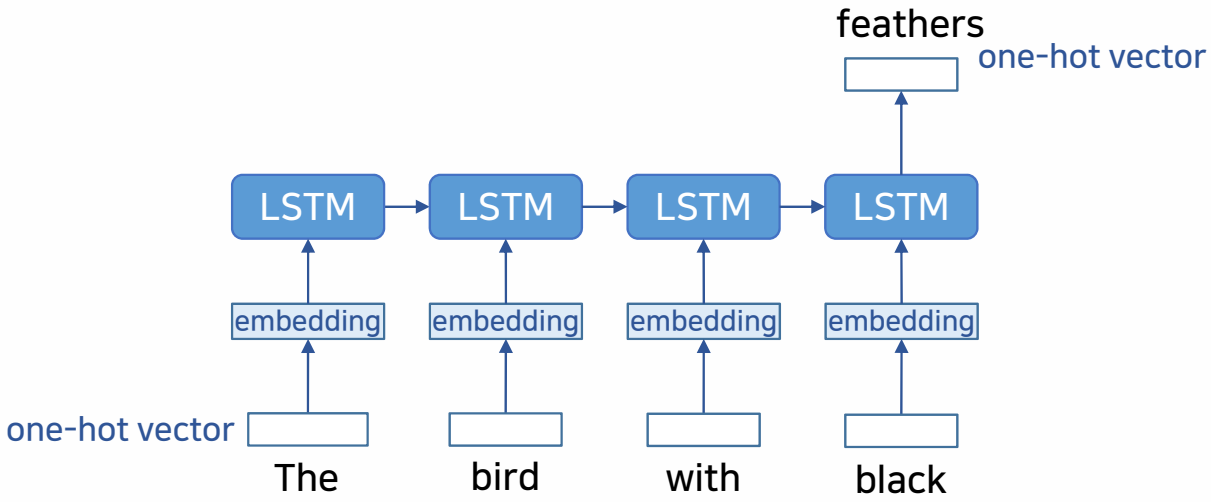
$(w_1, w_2, \cdots, w_n)$ → 언어 모델 → $P(w_1, w_2, \cdots, w_n)$

  - 단어의 시퀀스가 자연어 표현으로서 얼마나 적절한지를 평가하는 값
  - 구현 방법
    - ✓ 조건부확률 $\prod_{n=1}^{n} P(w_n|w_1, \cdots, w_{n-1})$을 이용하는 방법
    - ✓ 신경망(딥러닝)을 이용하는 방법 → 주로 RNN 사용

- 언어 모델의 활용
  - 문장 생성(요약, 번역 등) → P("나는 버스를 탄다") > P("나는 버스를 태운다")
  - 오타 교정 → P("빠르게 달려갔다") > P("빠르게 잘려갔다")

RNN 언어 모델

The bird with black feathers is on the tree.

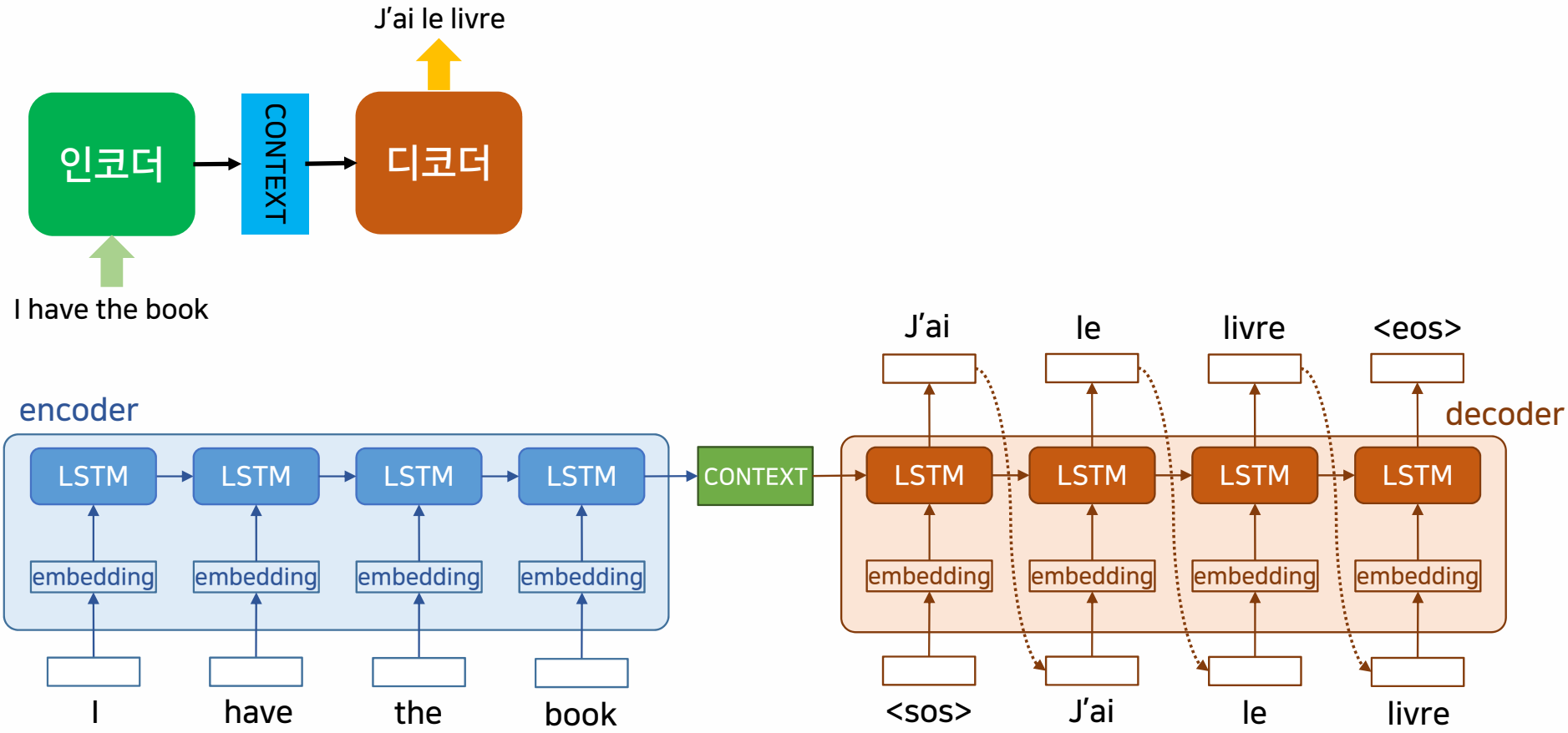


- 활용 → 텍스트 분류(감성 분류, 주제 분류 등), 기계번역 등

Seq2Seq 모델

- Sequence-to-Sequence Model
  - 입력된 시퀀스로부터 다른 도메인의 시퀀스를 출력
    - ✓ 응용 → 기계번역, 대화, 질의응답, 요약, STT(speech to text)
- 인코더와 디코더로 구성 → 각각 RNN(LSTM, GRU) 구조를 가짐
  - 인코더
    - ✓ 입력 → 임베딩 벡터로 표현된 단어의 시퀀스
    - ✓ 출력 → 입력 시퀀스를 하나의 벡터로 압축한 문맥 벡터 context vector
  - 디코더
    - ✓ 입력 → 인코더에서 출력된 문맥 벡터
    - ✓ 출력 → 단어 벡터를 순차적으로 출력

Seq2Seq 모델


J'ai le livre
인코더
CONTEXT
디코더
I have the book
encoder
LSTM
LSTM
LSTM
LSTM
embedding
embedding
embedding
embedding
I
have
the
book
CONTEXT
J'ai
le
livre
<eos>
decoder
LSTM
LSTM
LSTM
LSTM
embedding
embedding
embedding
embedding
<sos>
J'ai
le
livre

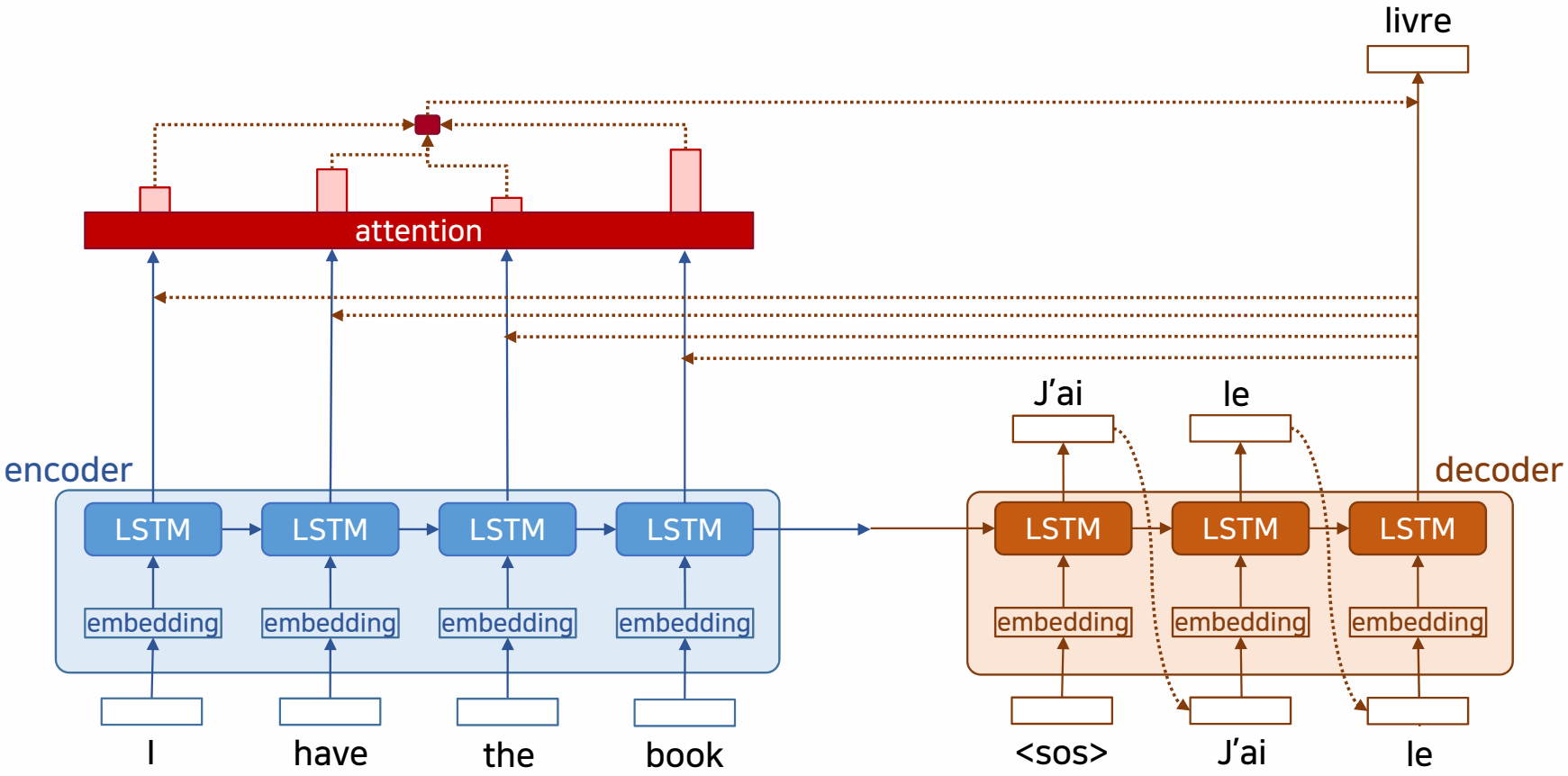
Attention 모델

- Seq2Seq 모델의 문제
  - 인코더로부터 얻어진 정보를 하나의 고정된 특징벡터로 요약/압축

    → 정보 손실
  - 입력 문장의 길이가 길어지면 성능 저하

- attention 모듈을 이용한 해결
  - 디코더에서 출력 단어를 생성할 때마다
    인코더의 전체 상태에 대한 선택적 주의를 통해 참조하는 방식

Attention 모델

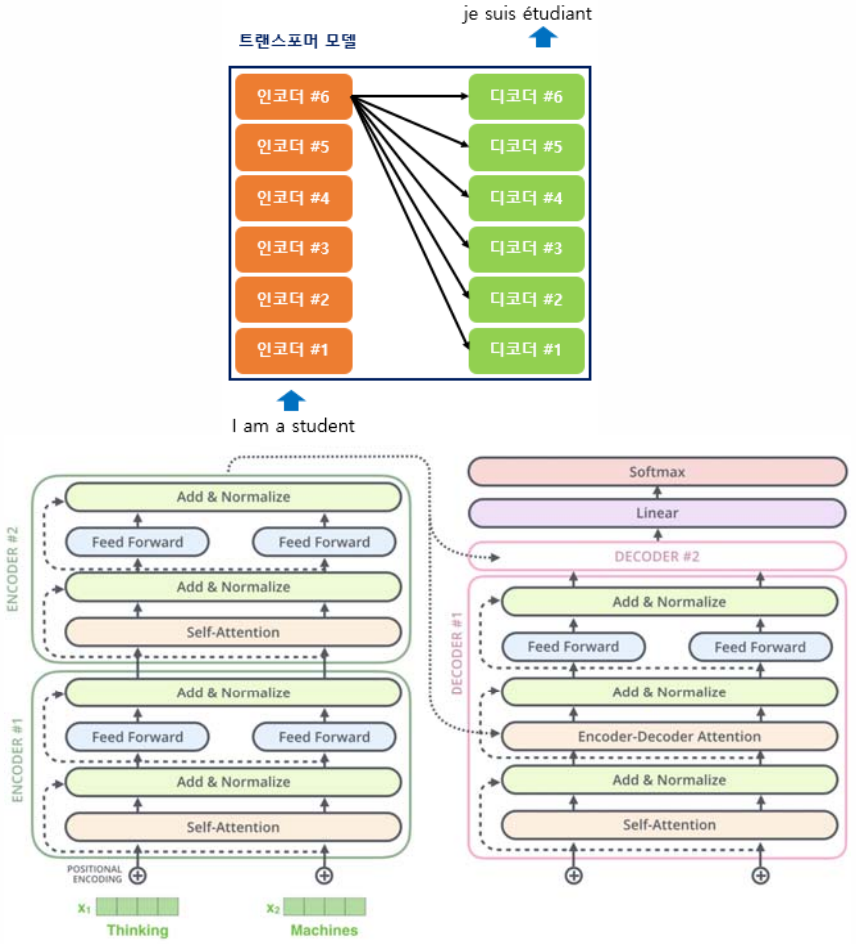


Transformer 모델

https://arxiv.org/abs/1706.03762
http://jalammar.github.io/illustrated-transformer/

Transformer

- Seq2Seq 모델의 인코더-디코더 구조 사용
- RNN 구조를 없애고
  여러 개의 인코더와 attention만으로 구현
- 인코더 → 한 번에 전체 시퀀스를 입력받음
  - ✓ 단어의 순서 정보를 나타내기 위해 positional encoding 사용
- 디코더 → 한 번에 하나씩 순차적으로 생성
- 빠른 학습, 우수한 성능

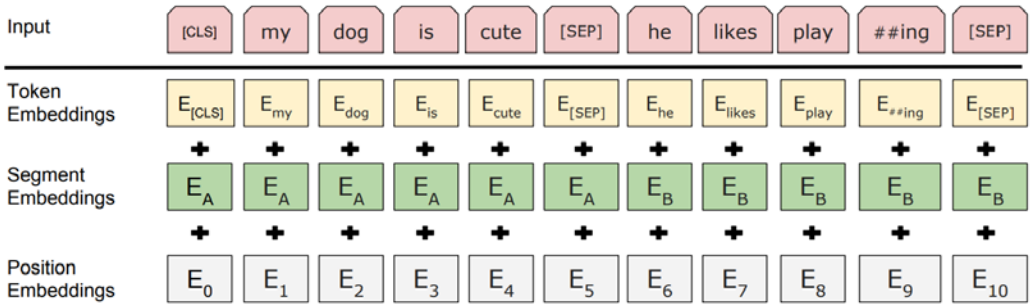


BERT

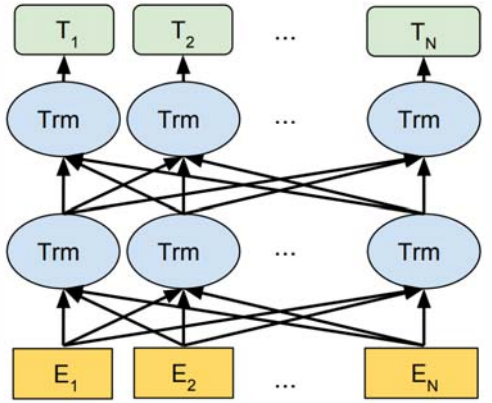
https://arxiv.org/pdf/1810.04805.pdf

BERT Bidirectional Encoder Representations from Transformers

- 구글에서 만든 새로운 언어 모델
- 3 종류의 입력 임베딩 수행



- 사전학습 pre-training 단계
  - ✓ 방대한 양의 데이터를 이용하여 학습한 언어 모델 구축
  - ✓ 책 말뭉치(800M 단어) + Wikipedia(2500M 단어)로 학습
- 미세조정 fine-tuning 단계
  - ✓ 사전에 학습된 모델을 특정 NLP 문제에 맞춰 추가 학습
  - ✓ 12개의 자연어처리 문제에 대해 최고 성능을 기록

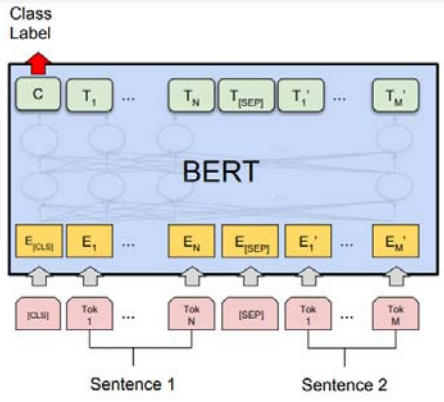


Transformer의 인코더 모델에 기반

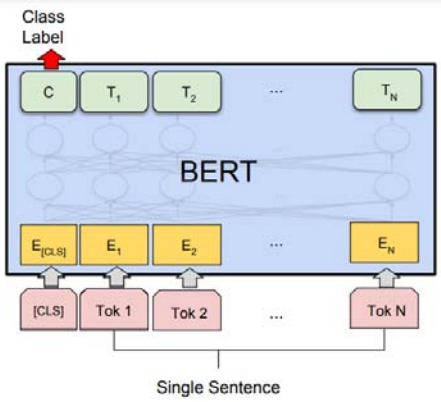
BERT

NLP 문제 유형에 따른 BERT 모델의 4가지 구성 방식

문장 간의 유사도 평가
질의-응답 문장의 적절성 평가



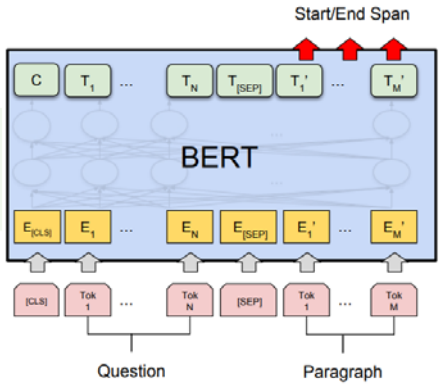
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



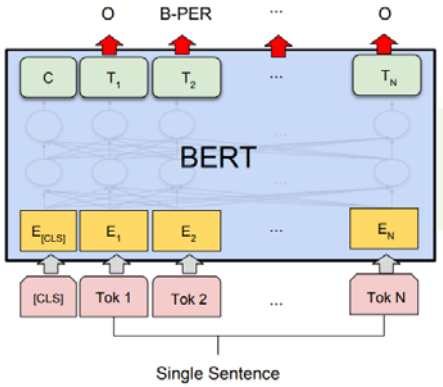
(b) Single Sentence Classification Tasks:
SST-2, CoLA

감성 분류, 기사 분류

하나의 질문/문단 → 응답 문장
질의응답

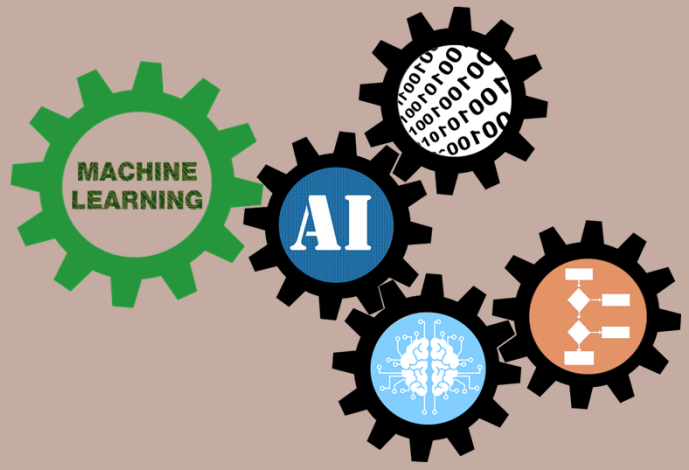


(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

문장의 각 단어에 대응되는 출력
품사 태깅



다음시간안내

제15강

강화학습