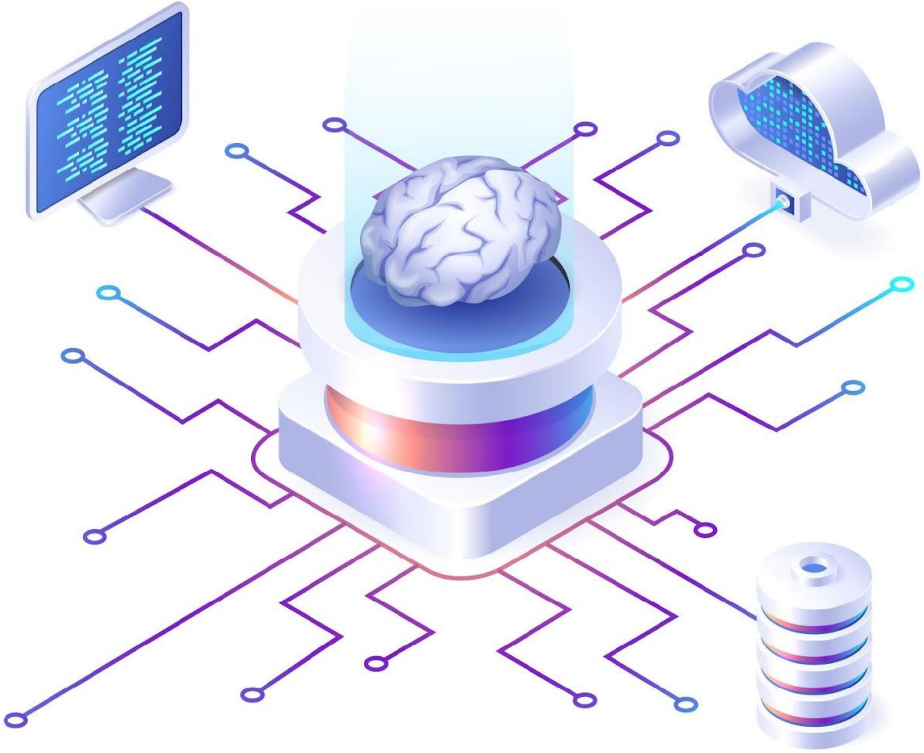
통계기반 자연어처리에서 딥러닝 자연어처리까지

실무형 인공지능 자연어처리



자연어 처리(NLP)란?

자연어 처리란?

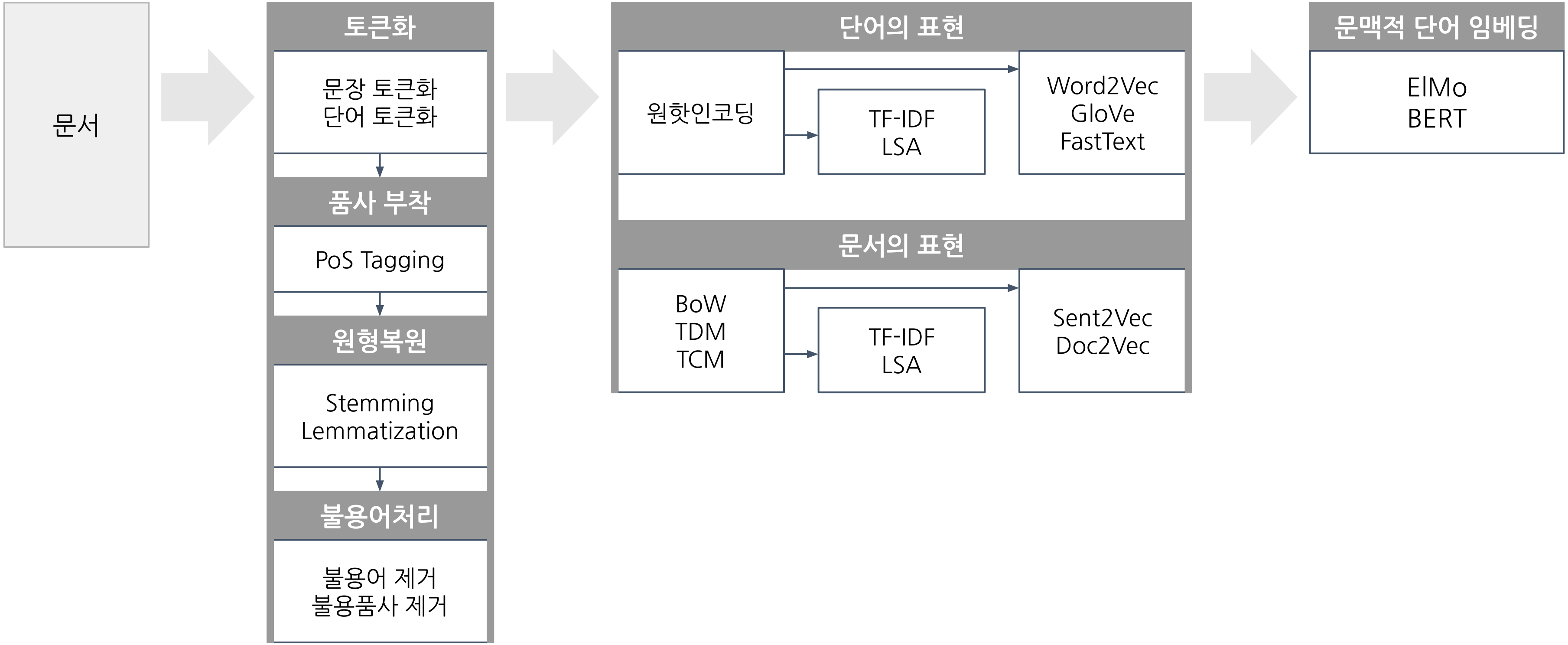
전통적인 프로그래밍 언어가 인간이 기계 언어로 기계(=컴퓨터)를 이해시키는 것이었다면,

<u>자연어 처리는 기계가 인간의 언어(=자연 언어)를 이해하여 소통하는 것을 말한다.</u>

통계기반 자연어 처리 절차

데이터 수집 단계	텍스트 전처리 단계	텍스트 분석 단계	시각화 단계
데이터 수집 Data Collection	토큰화 Tokenizing	주제어 찾기 Topic modeling	
데이터 정제 Data Cleansing	품사 부착 POS Tagging	문서 요약 Summarize	
	원형 복원 Stemming, Lemmatizing	문서 분류 Category classification	
	불용어 처리 Stopword	감성 분석 Sentiment analysis	

표현 (Representation)



1

단어의 표현
(Word Representation)

원핫-인코딩(One-Hot-Encoding)

원핫-인코딩은 단어(word)를 숫자로 표현하고자 할 때 적용할 수 있는 간단한 방법론

원숭이 ... ㅏ, 코끼리를

$$\text{원숭이} = [\overbrace{1,\ 0,\ 0}^{\text{차원의 수(예, 3차원)}}]$$

인덱스 인덱스 인덱스

원숭이 ...

바나나 ...

사과 = [0, 0, 1]

... 0, 0]

... 0, 0]

사과 = [0, 0, 1, 0]

코끼리 = [0, 0, 0, 1]

Insight campus SeSAC

TF-IDF (단어 빈도-역문서 빈도)

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

tf(d,t)	특정 문서 d에서의 특정 단어 t의 등장 횟수
df(t)	특정 단어 t가 등장한 문서의 수
idf(d, t)	df(t)의 역수

TF	IDF	TF-IDF	설명
높	높	높	특정 문서에 많이 등장하고 타 문서에 많이 등장하지 않는 단어 (중요 키워드)
높	낮	-	특정 문서에도 많이 등장하고 타 문서에도 많이 등장하는 단어
낮	높	-	특정 문서에는 많이 등장하지 않고 타 문서에만 많이 등장하는 단어
낮	낮	낮	특정 문서에 많이 등장하지 않고 타 문서에만 많이 등장하는 단어

LSA (잠재의미분석)

- TDM (문서-단어 행렬)은 sparse 함
- LSA를 활용하여 의미를 보존하며 밀집벡터(dense vector)를 생성할수 있음

2

문서의 표현

(Word Representation)

BoW (Bag of Word)

문서1: 오늘 동물원에서 코끼리를 봤어
문서2:오늘 동물원에서 원숭이에게 사과를 줬어

Step1. 각 토큰에 고유 인덱스 부여

Step2. 각 인덱스 위치에 토큰 등장 횟수를 기록

오늘	0
동물원에서	1
코끼리를	2
봤어	3
원숭이에게	4
사과를	5
줬어	6

	오늘	동물원에서	코끼리를	봤어	원숭이에게	사과를	줬어
문서1	1	1	1	1	0	0	0

	오늘	동물원에서	코끼리를	봤어	원숭이에게	사과를	줬어
문서2	1	1	0	0	1	1	1

Insight campus SeSAC

TDM (단어-문서 행렬)

BoW(Bag of Words) 중 하나
문서에 등장하는 각 단어 빈도를 행렬로 표현한 것

문서1: 동물원 코끼리
문서2: 동물원 원숭이 바나나
문서3: 엄마 코끼리 아기 코끼리
문서4: 원숭이 바나나 코끼리 바나나

	동물원	코끼리	원숭이	바나나	엄마	아기
문서1	1	1	0	0	0	0
문서2	1	0	1	1	0	0
문서3	0	2	0	0	1	1
문서4	0	1	1	2	0	0

TF-IDF (단어 빈도-역문서 빈도)

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

tf(d,t)	특정 문서 d에서의 특정 단어 t의 등장 횟수
df(t)	특정 단어 t가 등장한 문서의 수
idf(d, t)	df(t)의 역수

TF	IDF	TF-IDF	설명
높	높	높	특정 문서에 많이 등장하고 타 문서에 많이 등장하지 않는 단어 (중요 키워드)
높	낮	-	특정 문서에도 많이 등장하고 타 문서에도 많이 등장하는 단어
낮	높	-	특정 문서에는 많이 등장하지 않고 타 문서에만 많이 등장하는 단어
낮	낮	낮	특정 문서에 많이 등장하지 않고 타 문서에만 많이 등장하는 단어

LSA (잠재의미분석)

- TDM (문서-단어 행렬)은 sparse 함
- LSA를 활용하여 의미를 보존하며 밀집벡터(dense vector)를 생성할수 있음

	단어1	단어2	단어3	단어4
문서1				
문서2	문서 내 단어 등장 빈도			
문서3				
문서4				

문서-단어행렬

=

	차원1	차원2
문서1		
문서2	문서별 주제 가중치	
문서3		
문서4		

문서벡터행렬

✖

	차원1	차원2
차원1	특이값	
차원2		

✖

	단어1	단어2	단어3	단어4
차원1	단어별 주제 가중치			
차원2				

단어벡터행렬

	차원1	차원2
문서1	문서1 벡터	
문서2	문서2 벡터	
문서3	문서3 벡터	
문서4	문서4 벡터	

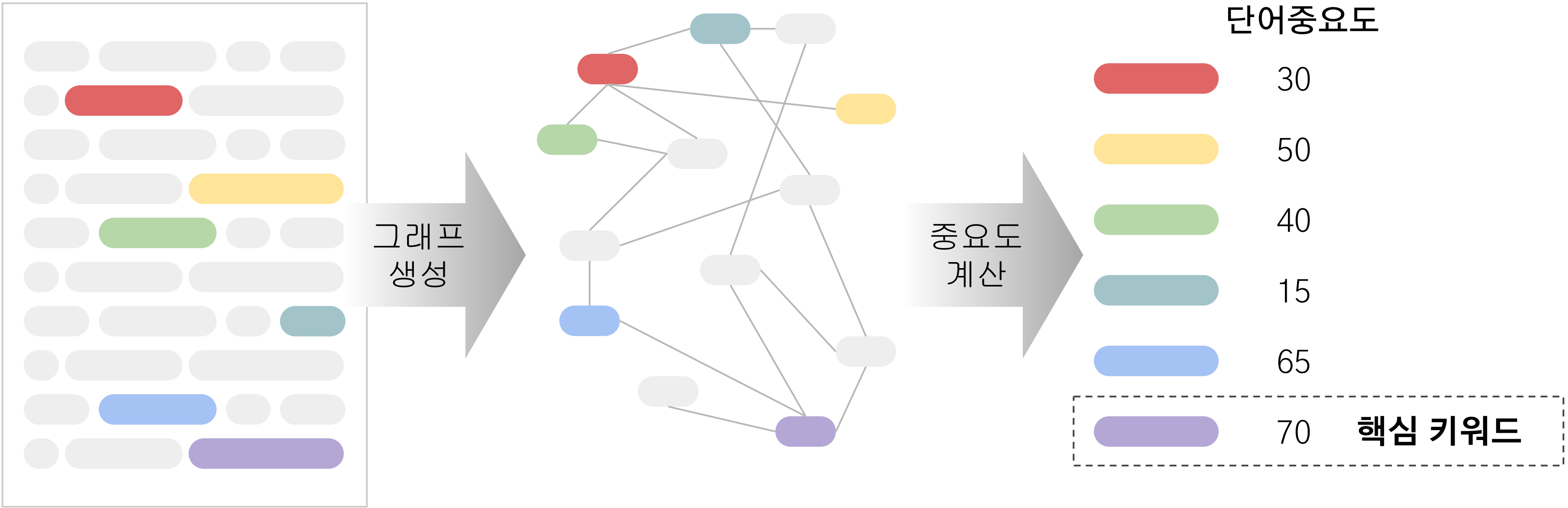
	단어1	단어2	단어3	단어4
차원1	단어1 벡터	단어2 벡터	단어3 벡터	단어4 벡터
차원2				

3

키워드 추출
(Keyword Extraction)

Insight campus SeSAC

TextRank



Insight campus SeSAC

TextRank

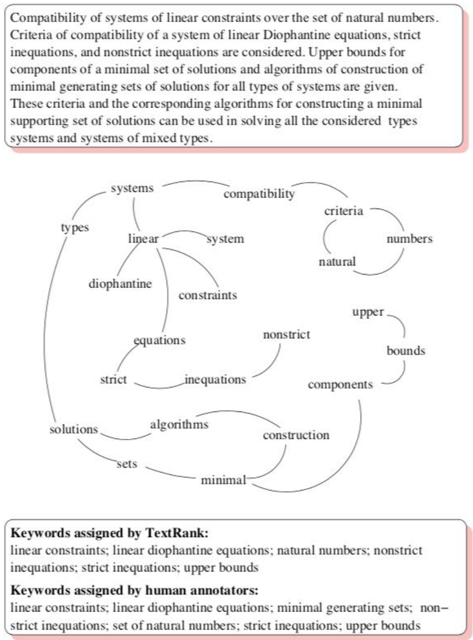


Figure 2: Sample graph build for keyphrase extraction from an *Inspec* abstract

The TextRank keyword extraction algorithm is fully unsupervised, and proceeds as follows. First,the text is tokenized, and annotated with part of speech tags – a preprocessing step required to enable the application of syntactic filters… Next, all lexical units that pass the syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of words. After the graph is constructed (undirected unweighted graph), the score associated with each vertex is set to an initial value of 1, and the ranking algorithm described in section 2 is run on the graph for several iterations until it converges– usually for 20-30 iterations, at a threshold of 0.0001.… For this example, the lexical units found to have higher "importance" by the TextRank algorithm are (with the TextRank score indicated in parenthesis): numbers (1.46), inequations (1.45), linear (1.29), dio phantine (1.28), upper (0.99), bounds (0.99), strict (0.77)

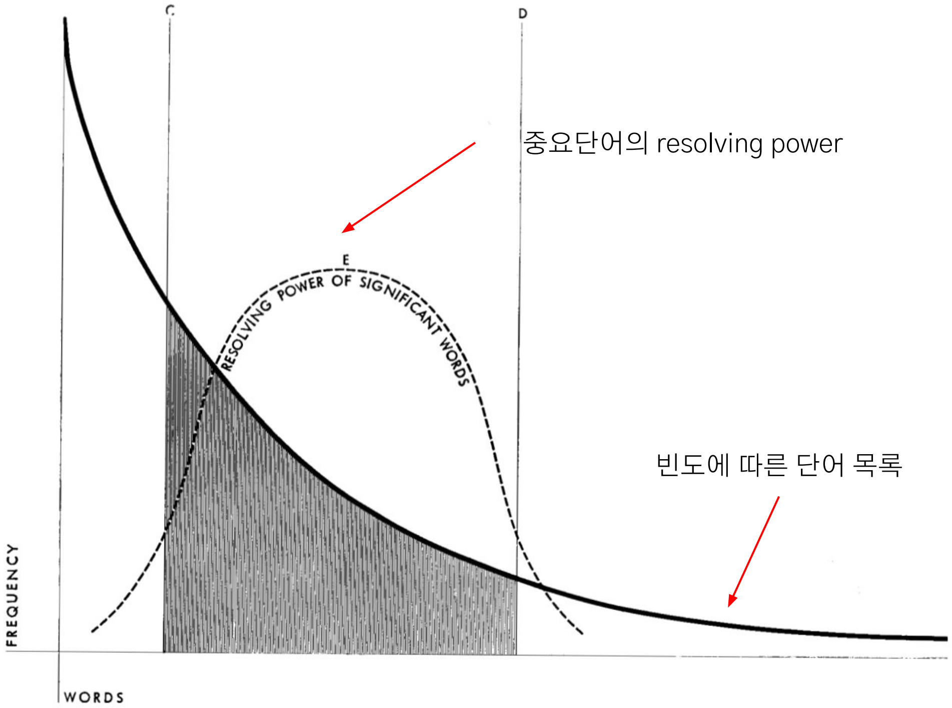
- 1단계 : 텍스트는 품사가 태깅되어 토큰화 됨
- 2단계 : 단어 윈도(window of words)에 동시 등장한 토큰 사이는 엣지를 추가하여 그래프를 생성
- 3단계 : 0.0001을 threshold로 20-30회 반복

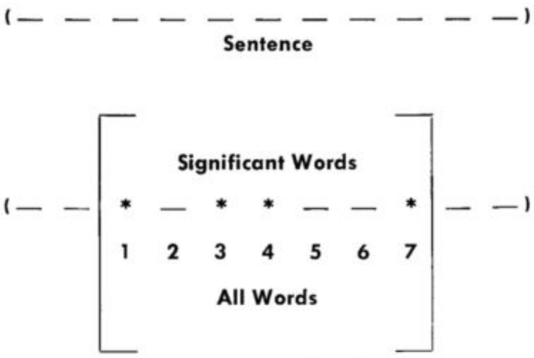
4 문서 요약

(Document Summarization)

Luhn Summerizer

Figure 1 **Word-frequency diagram.**
Abscissa represents individual words arranged in order of frequency.



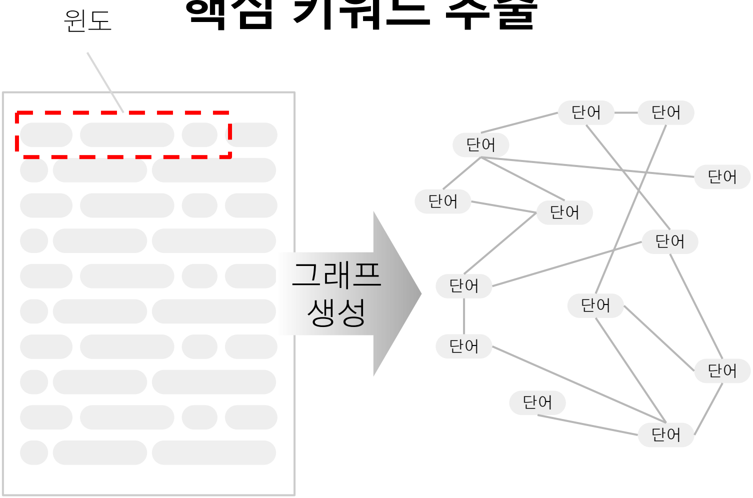


Portion of sentence bracketed by and including significant words not more than four non-significant words apart. If eligible, the whole sentence is cited.

Figure 2 **Computation of significance factor.**
The square of the number of bracketed significant words (4) divided by the total number of bracketed words (7) = 2.3.

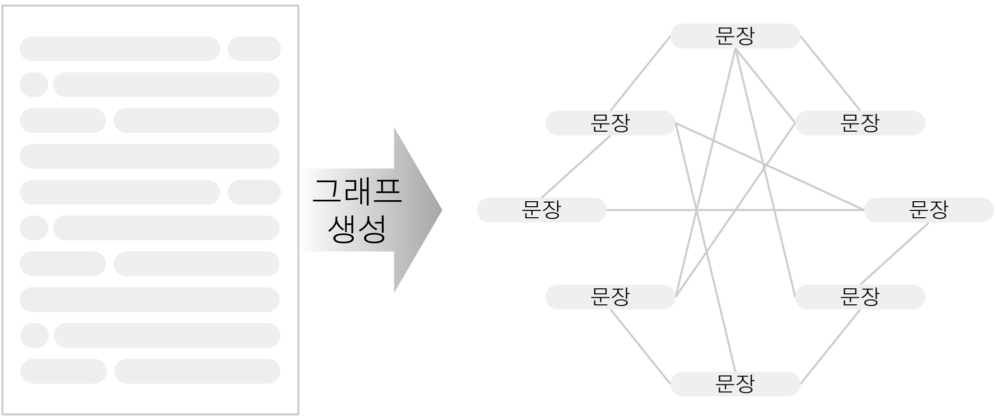
키워드 추출 vs 문서요약

핵심 키워드 추출



윈도가 이동하며
그래프 생성

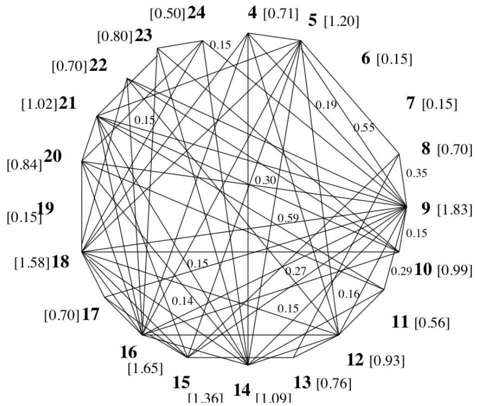
문서 요약



모든 문장간 유사도를 기준으로
그래프 생성

TextRank

3: BC–HurricaineGilbert, 09–11 339
4: BC–Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

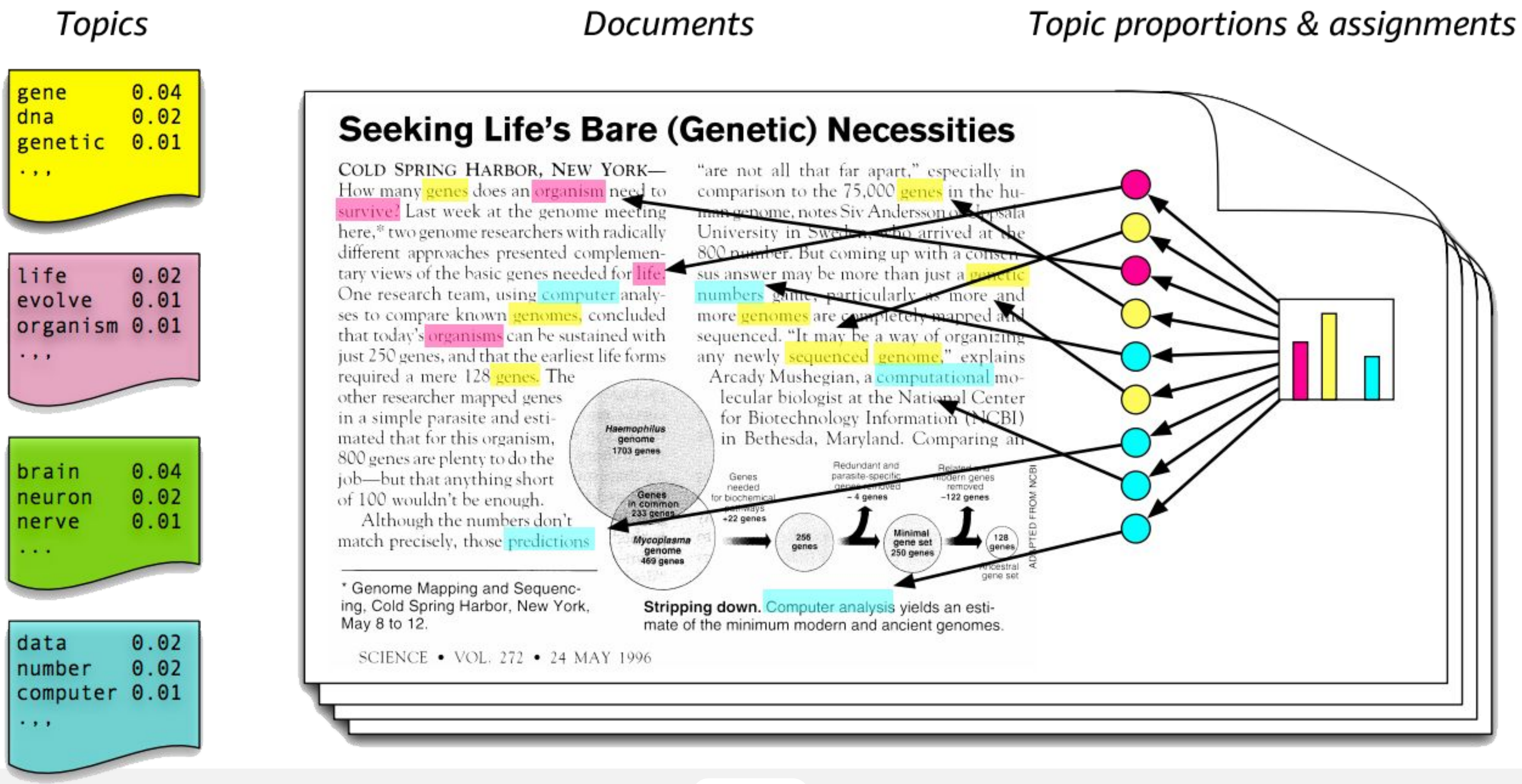


TextRank extractive summary
Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast.

Manual abstract I
Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high wind and seas. Tropical storm Gilbert formed in the eastern Carribean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and in the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and down–graded to a tropical storm.

Manual abstract II
Tropical storm Gilbert in the eastern Carribean strenghtened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15 mph with a broad area of cloudiness and heavy weather with sustained winds of 75 mph gusting to 92 mph. The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

5 토픽 모델링
(Topic Modeling)


Topics
Documents
Topic proportions & assignments
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Seeking Life's Bare (Genetic) Necessities
COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.
Although the numbers don't match precisely, those predictions
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed −4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
SCIENCE • VOL. 272 • 24 MAY 1996

LSA (잠재의미분석)

- 토픽을 추출된 키워드들의 분포로 나타냄으로써 텍스트 내의 구조를 파악

문서-단어행렬
m x n = 6 x 4

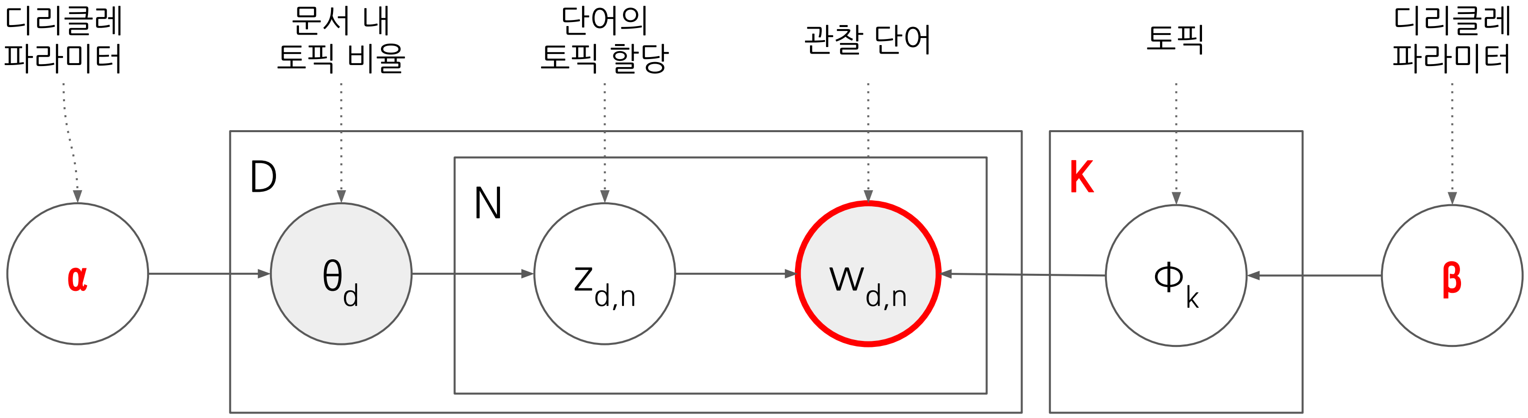
좌특이벡터 (문서벡터)
m x m = 6 x 6

특이값
m x n = 6 x 4

우특이벡터 (단어벡터)
n x n = 4 x 4

LDA (잠재 디리클레 할당 모델)

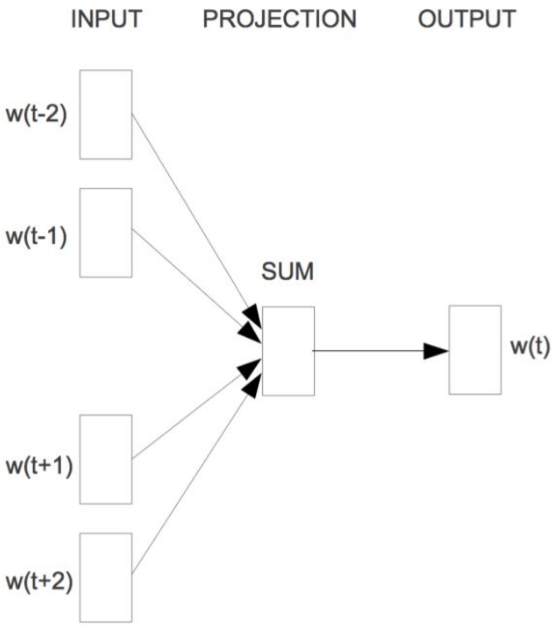
α	디리클레 파라미터 (보통 0.1)	D	전체 문서 갯수
θ_d	문서 내 토픽 비율	Φ_k	토픽
$z_{d,n}$	단어의 토픽 할당	K	토픽수
$w_{d,n}$	관찰단어	β	토픽 하이퍼파라미터 (보통 0.001)
N	N은 d번째 문서의 단어 수		



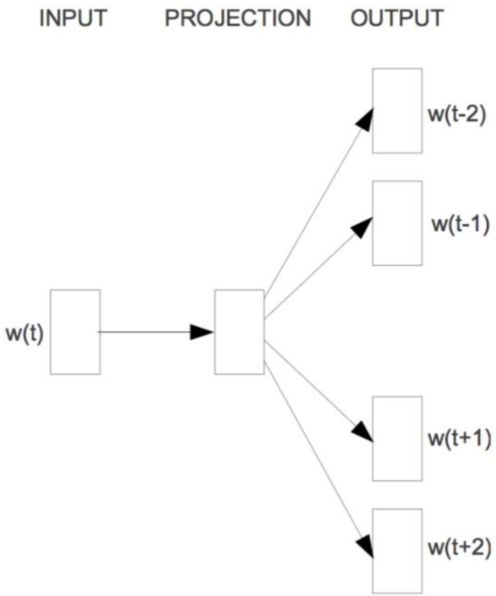
1

단어 임베딩
(Word Embedding)

Word2Vec



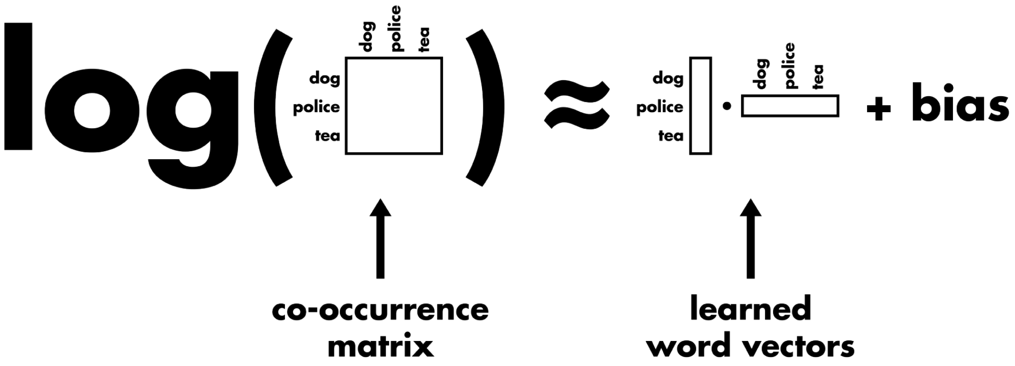
CBOW



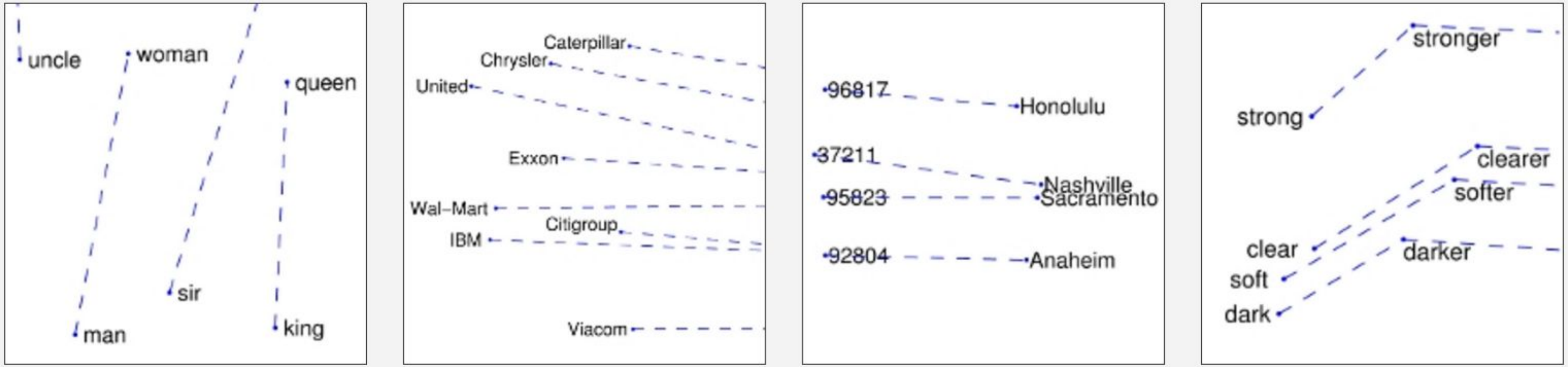
Skip-gram

GloVe

- 임베딩된 두 단어벡터의 내적이 말뭉치 전체에서의 동시 등장확률 로그값이 되도록 목적함수를 정의 (their dot product equals the logarithm of the words' probability of co-occurrence)
- 특정 단어 k가 주어졌을 때 임베딩된 두 단어벡터의 내적이 두 단어의 동시등장확률 간 비율이 되도록 임베딩
    - solid라는 단어가 주어졌을 때 ice와 steam 벡터 사이의 내적값이 8.9가 되도록
    - gas가 주어졌을 때 ice와 steam 벡터 사이의 내적값이 0.0085가 되도록



GloVe의 결과

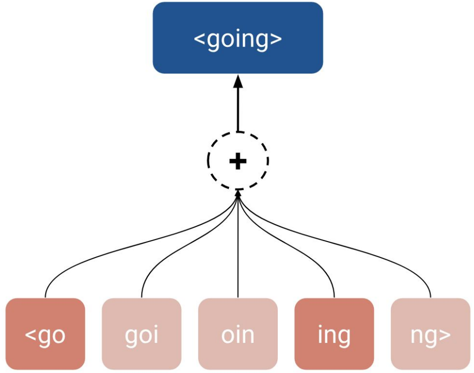


man - woman company - ceo city - zip code comparative - superlative

FastText

- FastText에서 각 단어를 글자의 n-gram으로 나타냄
- 예를 들어, tri-gram의 경우, apple은 app, ppl, ple로 분리하고 임베딩
- FastText에서 birthplace(출생지)란 단어를 학습하지 않은 상태라고 해보자.
  - 다른 단어 n-gram으로서 birth와 place를 학습한 적이 있다면 birthplace의 임베딩 벡터 (Embedding Vector)를 만들어낼 수 있음

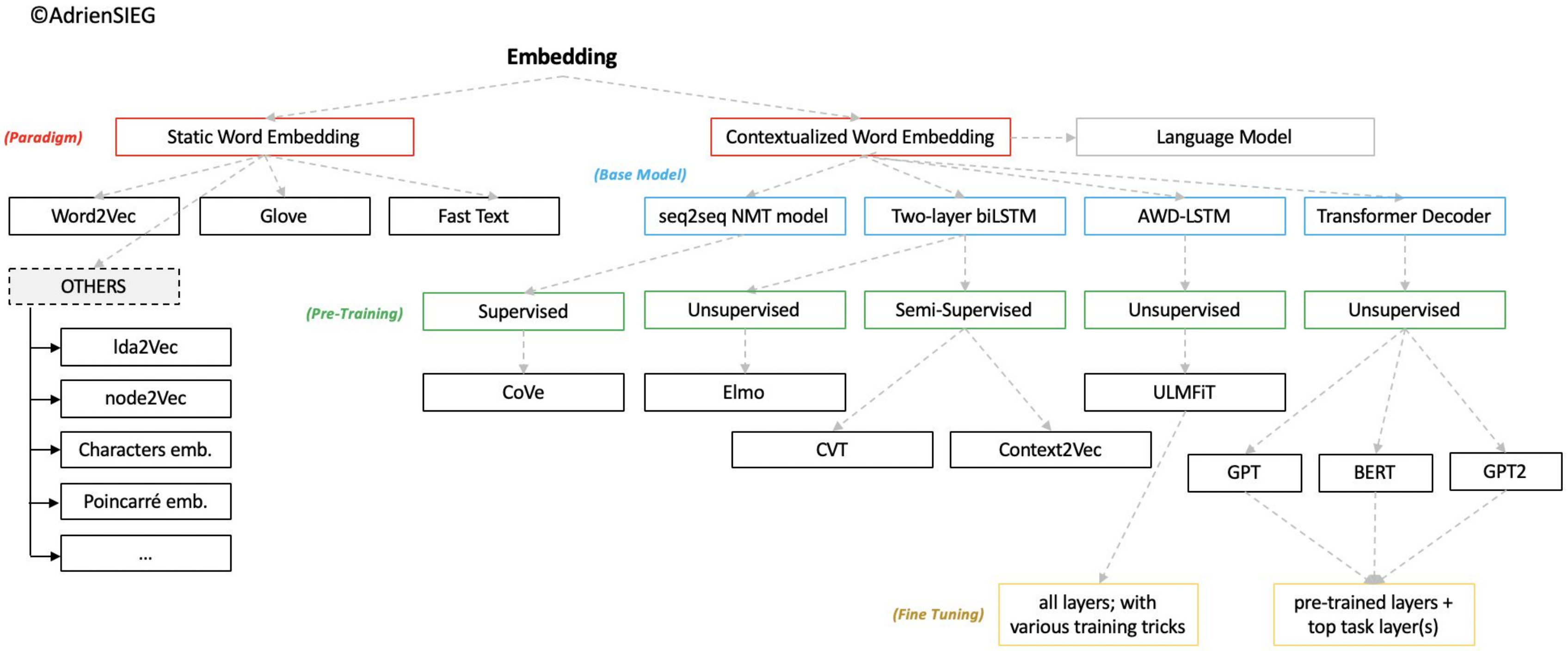
〈ap, app, ppl, ple, le〉 # n = 3 이므로 길이가 3
〈apple〉 # 특별 토큰



character 3-grams

2

문맥적 단어 임베딩
(Contextualized Word Embedding)



©AdrienSIEG
Embedding
(Paradigm)
Static Word Embedding
Contextualized Word Embedding
Language Model
(Base Model)
Word2Vec
Glove
Fast Text
seq2seq NMT model
Two-layer biLSTM
AWD-LSTM
Transformer Decoder
OTHERS
lda2Vec
node2Vec
Characters emb.
Poincarré emb.
...
(Pre-Training)
Supervised
Unsupervised
Semi-Supervised
Unsupervised
Unsupervised
CoVe
Elmo
ULMFiT
CVT
Context2Vec
GPT
BERT
GPT2
(Fine Tuning)
all layers; with various training tricks
pre-trained layers + top task layer(s)

BERT 까지

단어의 표현

문맥의 표현

RNN

- 순환 구조로 시계열에 대한 고려 가능
- 장기문맥 손실 (Long dependency)

LSTM / GRU

- grad vanishing/exploding 문제 해결
- 장기문맥에 대한 고려

ElMo

- biLM (양방향 언어모델)
- Bidirectional, Stacking

CNN

- 이미지 분류에 많이 사용
- 시계열에 대한 고려 X

Seq2Seq

- Encoder – Decoder 구조
- Many to many
- 고정 문맥 벡터로 표현하여 정보 손실 발생
- Teacher forcing

Seq2Seq with Attention

- 집중해야할 정보를 고려 = attention
- LSTM 기반 attention
- Attention mechanism

Transformer
(Attention is all you need)

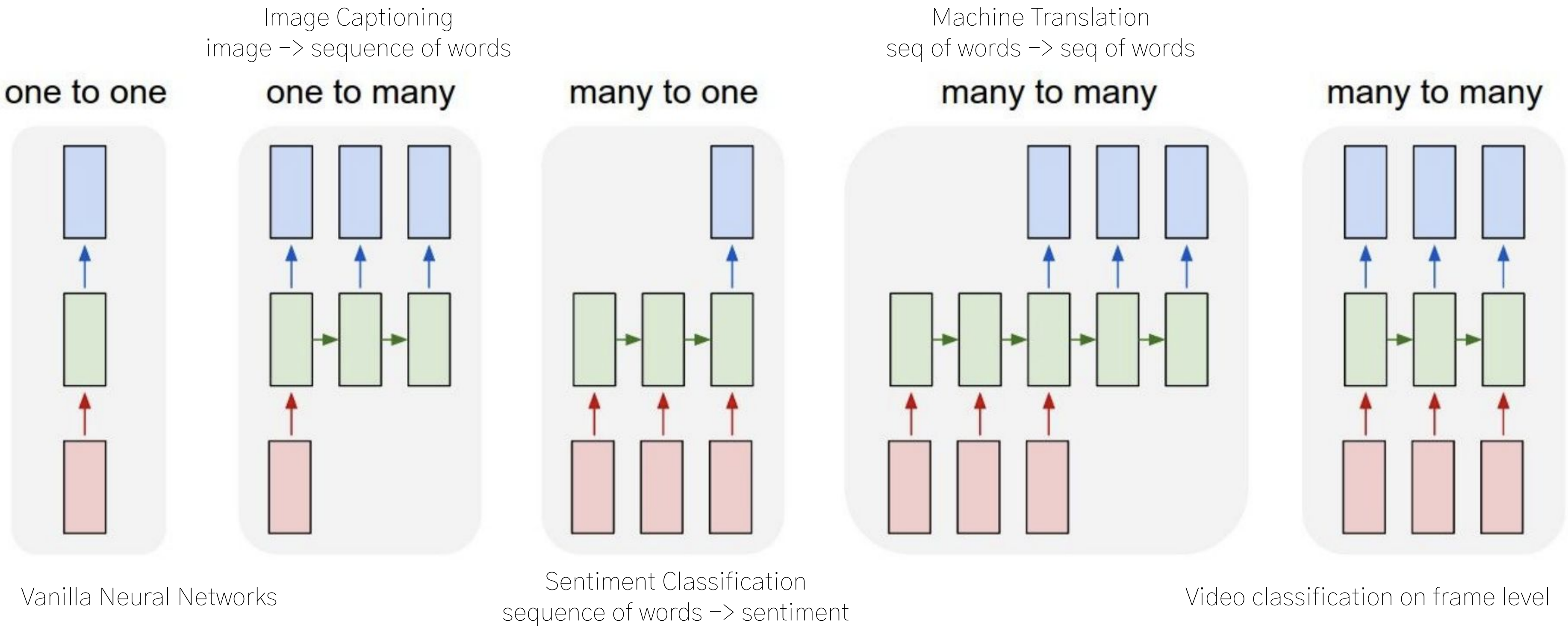
- LSTM을 제거. 속도향상
- Self-attention
- Multihead attention
- Positional encoding

BERT

- Wordpiece tokenizer
- Masked Language Model(MLM)
- Next Sentence Prediction (NSP)
- Position embedding
- Pre-trained model
- Fine tuning

1 RNN & ElMo & CNN

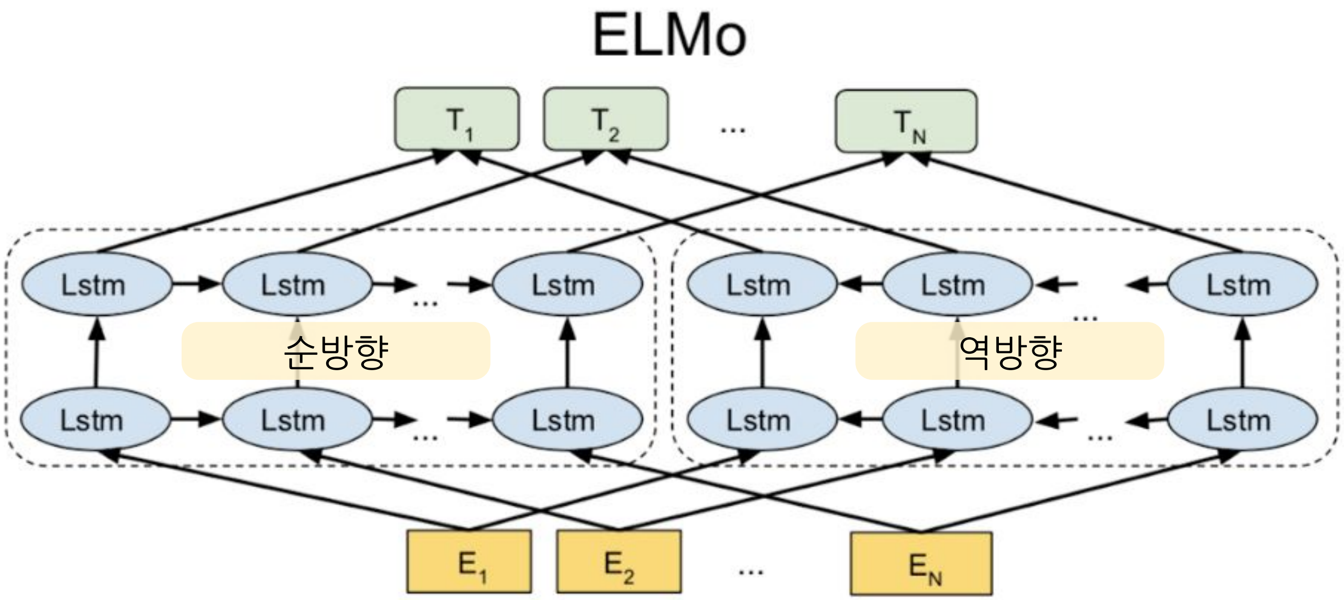
RNN



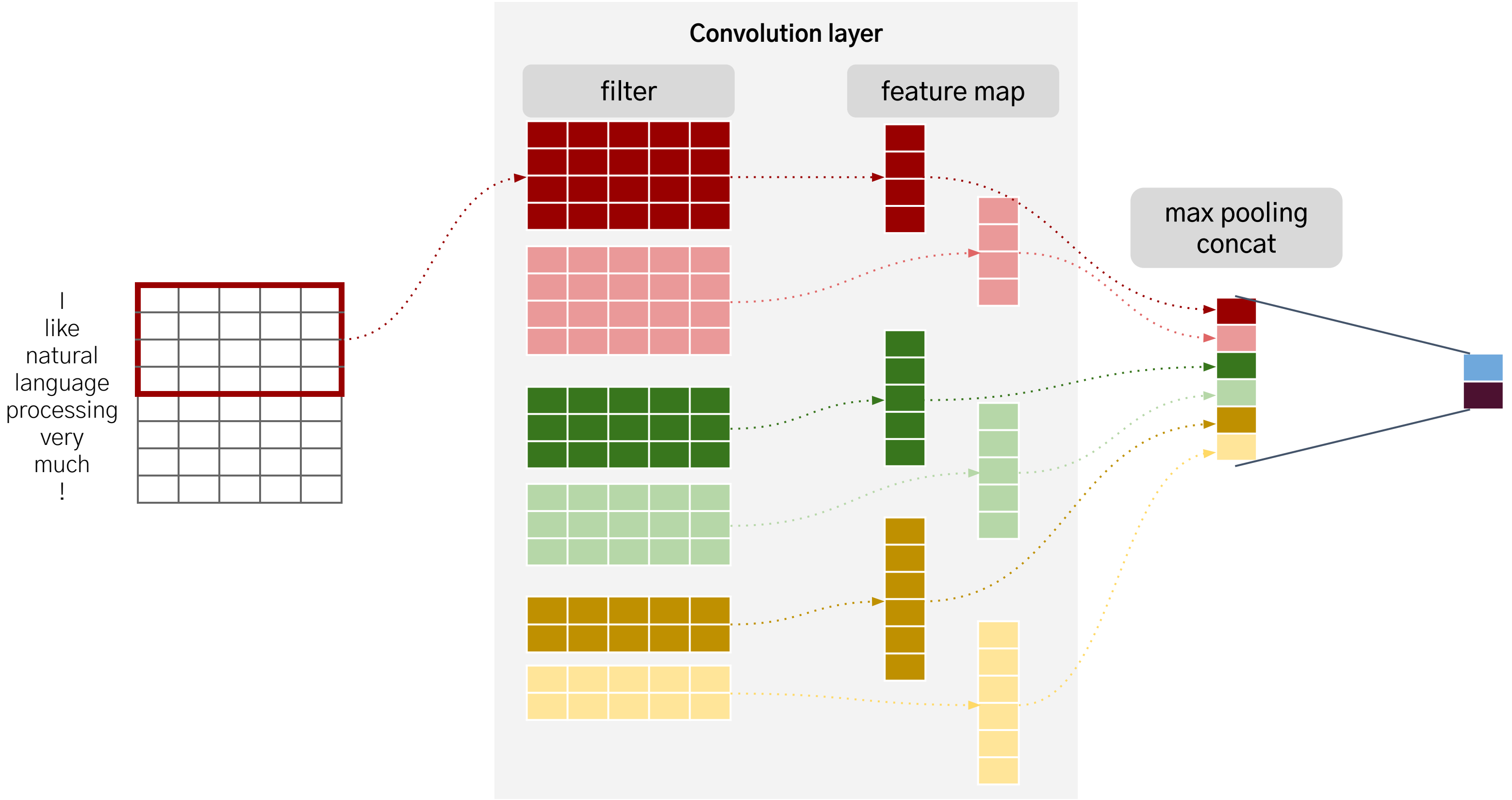
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

ELMo

- RNN으로 단어를 예측하는 것은 문맥을 고려한 단어 예측
- ELMo는 순방향 / 역방향으로 예측하는 biLM으로 사전 훈련

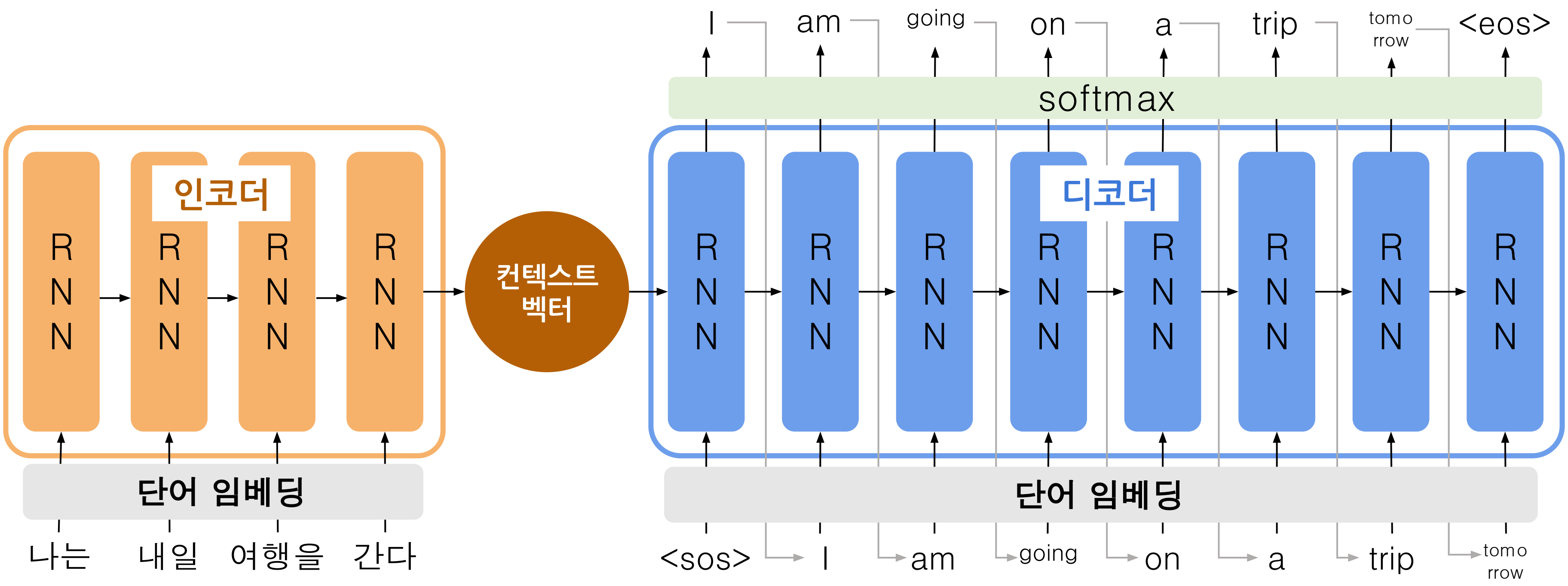


CNN


I
like
natural
language
processing
very
much
!
Convolution layer
filter
feature map
max pooling
concat

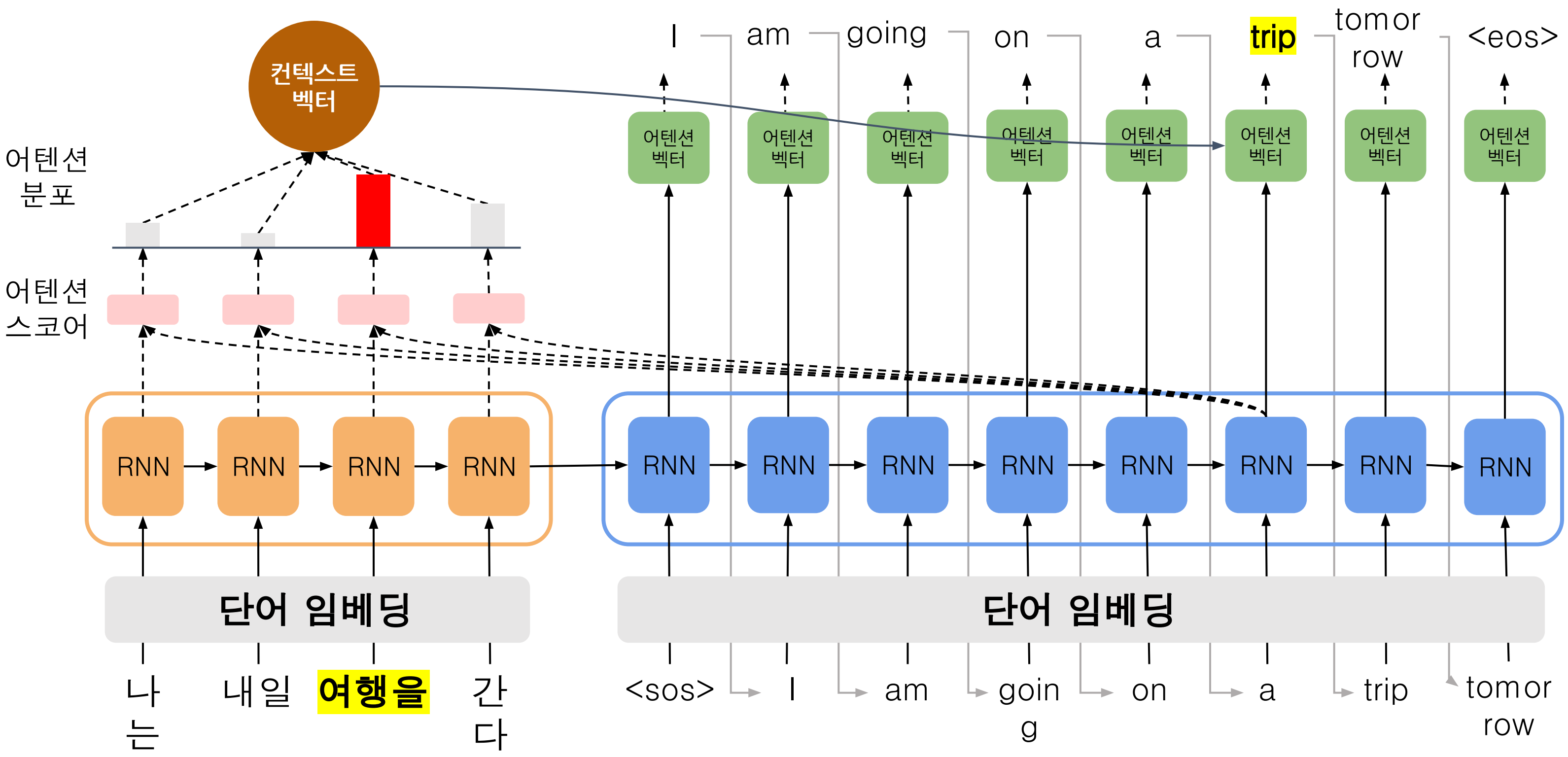
2 Seq2Seq & Attention

Seq2Seq



인코더
R N N
R N N
R N N
R N N
단어 임베딩
나는
내일
여행을
간다
컨텍스트 벡터
디코더
softmax
I
am
going
on
a
trip
tomo rrow
<eos>
R N N
단어 임베딩
<sos>
I
am
going
on
a
trip
tomo rrow

Seq2Seq with Attention



3 Transformer & BERT

Transformer

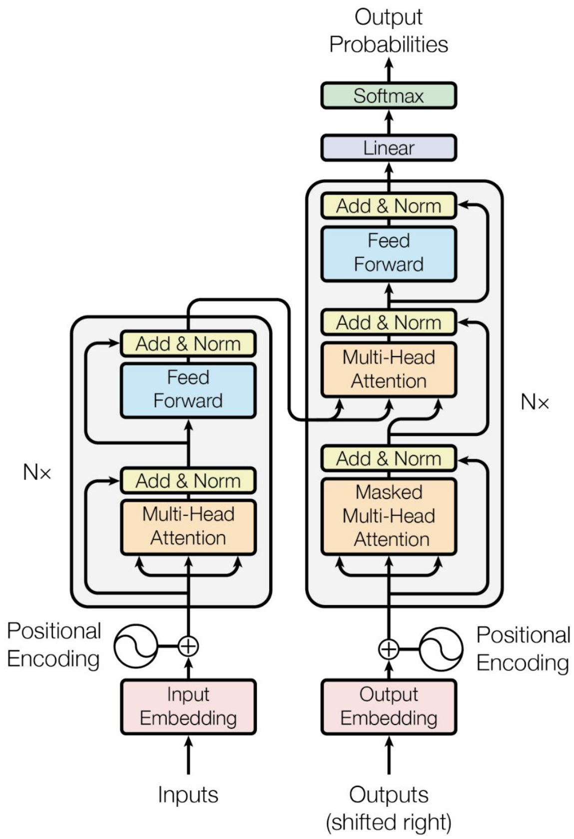
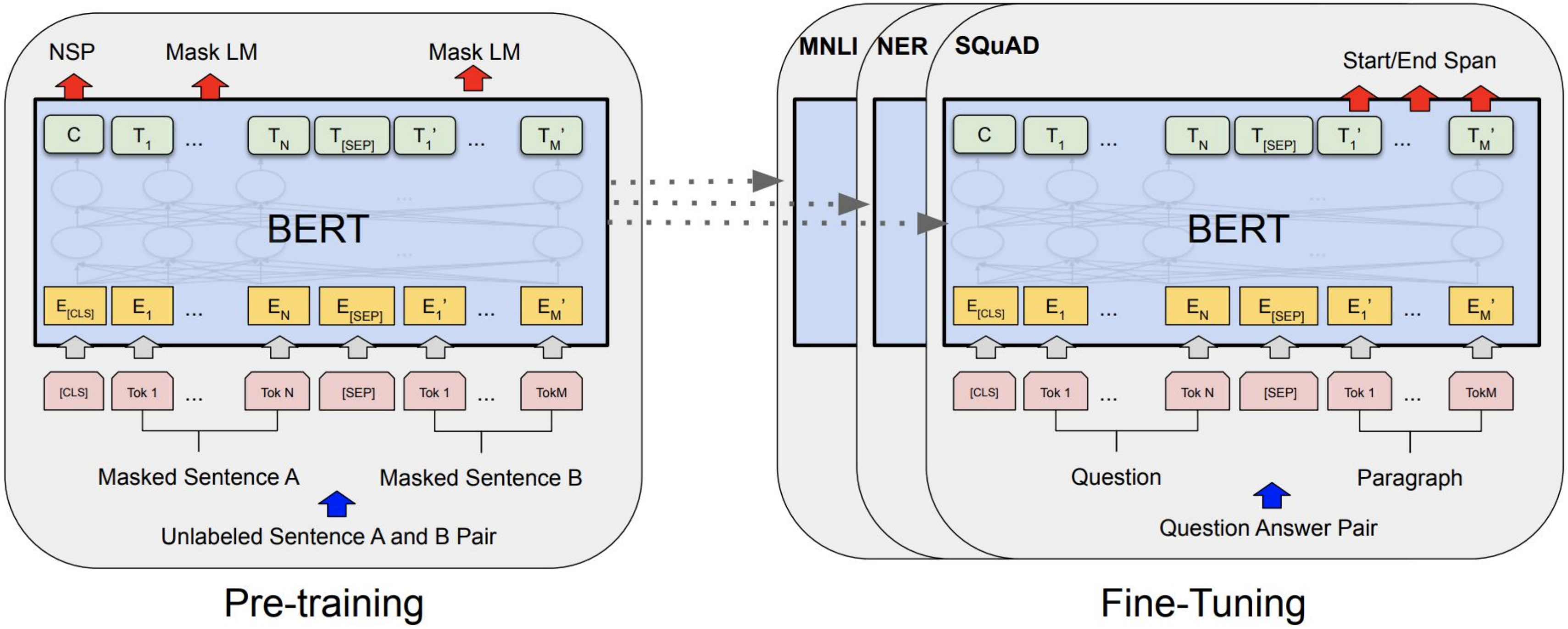


Figure 1: The Transformer - model architecture.

BERT



NSP
Mask LM
Mask LM
C
T1
TN
T[SEP]
T1'
TM'
BERT
E[CLS]
E1
EN
E[SEP]
E1'
EM'
[CLS]
Tok 1
Tok N
[SEP]
Tok 1
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
C
T1
TN
T[SEP]
T1'
TM'
BERT
E[CLS]
E1
EN
E[SEP]
E1'
EM'
[CLS]
Tok 1
Tok N
[SEP]
Tok 1
TokM
Question
Paragraph
Question Answer Pair
Fine-Tuning

감사합니다.

daniel@fins.ai