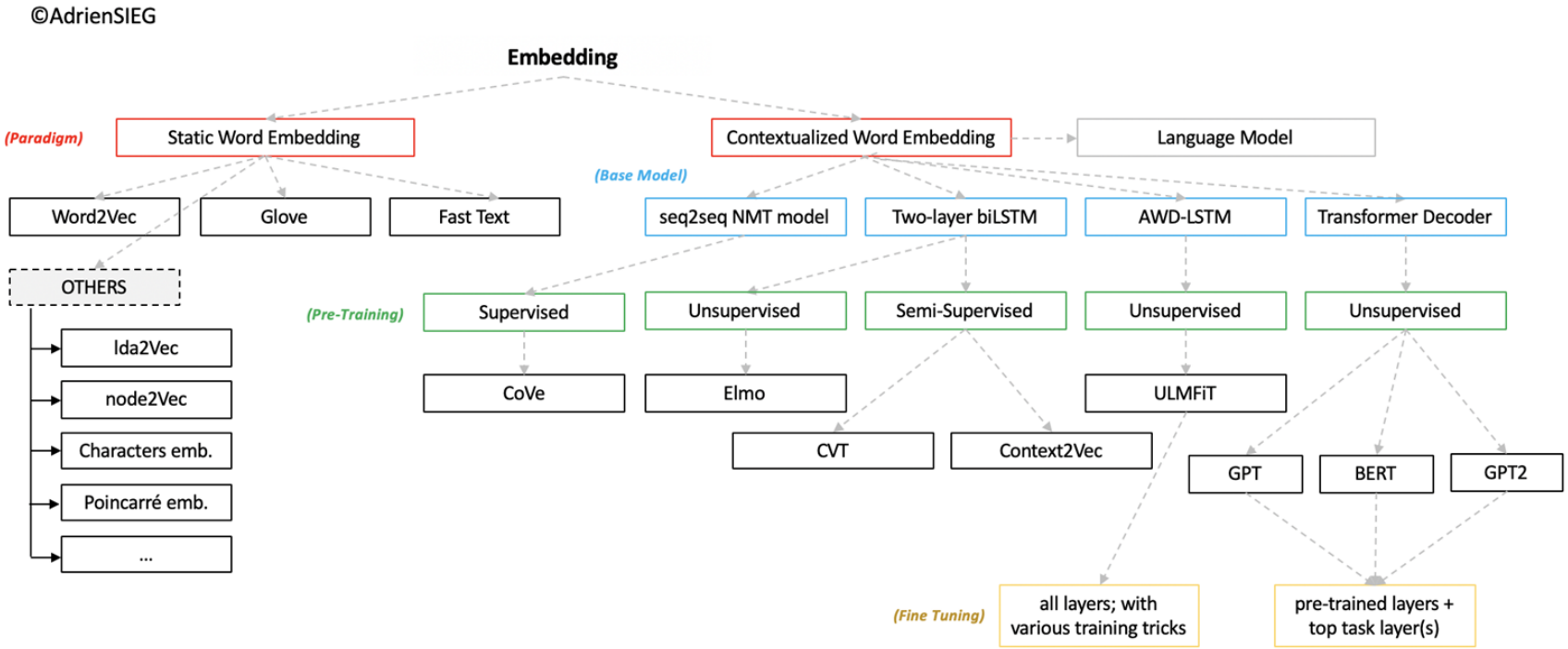
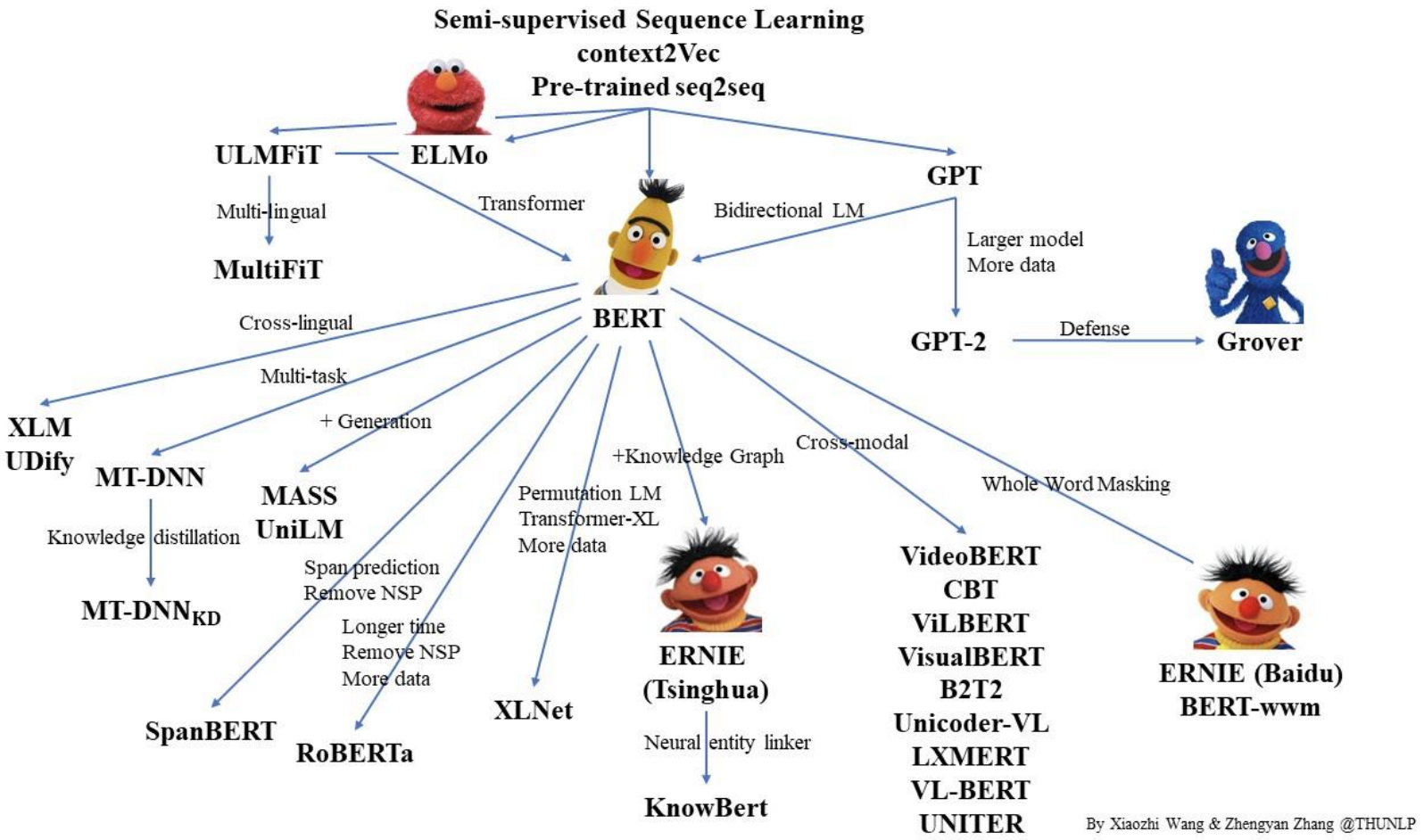
문맥적 단어 임베딩 (Contextualized Embedding)

자연어처리 텍스트마이닝



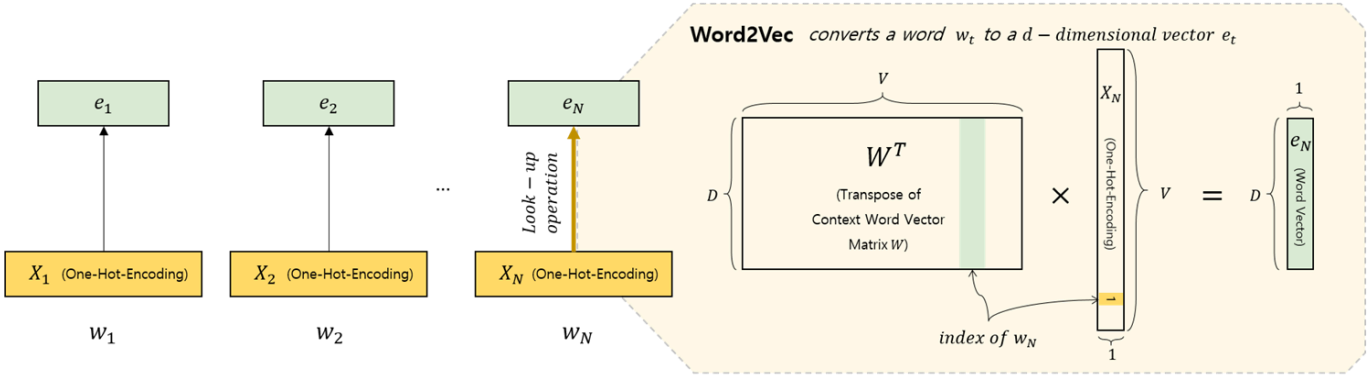


문맥적 단어 임베딩 (Contextualized Word Embedding)

- Contextualized Word Embedding은 문맥에 따라 vector를 생성
- 같은 단어여도 문맥에 따라 다른 vector가 생성될 수 있음
- 대표적으로 ELMo, BERT, OpenAI GPT
- 이들의 특징은 같은 단어라도 문맥에 따라 다른 방식으로 표현(representation)

문맥적 단어 임베딩 (Contextualized Word Embedding)

- Word2Vec의 Embedding은 단어 단위
- 각 단어의 one-hot-encoding vector가 W_T와 곱해져서 Word Vector를 얻게됨

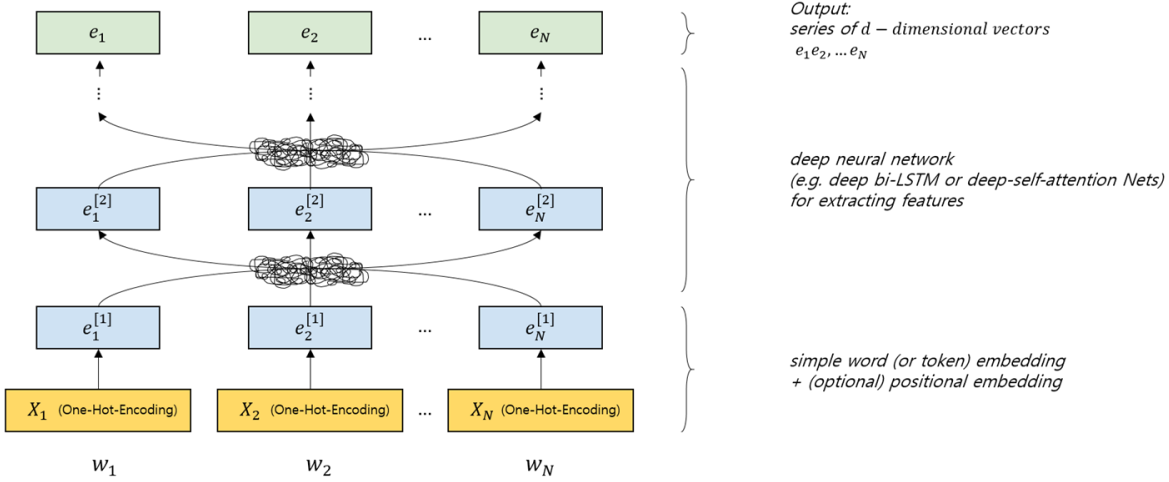


문맥적 단어 임베딩 (Contextualized Word Embedding)

- Contextualized Word Representation의 경우 문장을 입력 받아 각 단어에 대한 representation을 산출
- ‘문맥’에 의존적인 ‘단어의 의미’를 잡아내는 feature가 산출

Contextualized Word Representations

converts a sequence of words (more precisely, tokens) $w_1 w_2 \ldots w_N$ *to a series of* $d-$ *dimensional vectors* $e_1 e_2, \ldots e_N$



문맥적 단어 임베딩 (Contextualized Word Embedding)

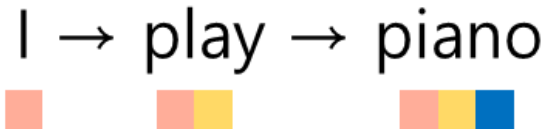
- Contextualized Word Representation의 경우 문장을 입력 받아 각 단어에 대한 representation을 산출
- ‘문맥’에 의존적인 ‘단어의 의미’를 잡아내는 feature가 산출

관점	Word2Vec	Contextualized Word Representation
Input	단어 단위	문장 단위 (단어의 시퀀스)
Layer	(일반적으로) 단층	(일반적으로) 다계층
Output	해당 단어에 대한 Embedding	문장을 구성하는 각 단어에 대한 Embedding들

문맥적 단어 임베딩 (Contextualized Word Embedding)

- *단순한 단어 임베딩 예시*

- *순환 신경망 언어 모델을 활용한 단어 임베딩 예시*

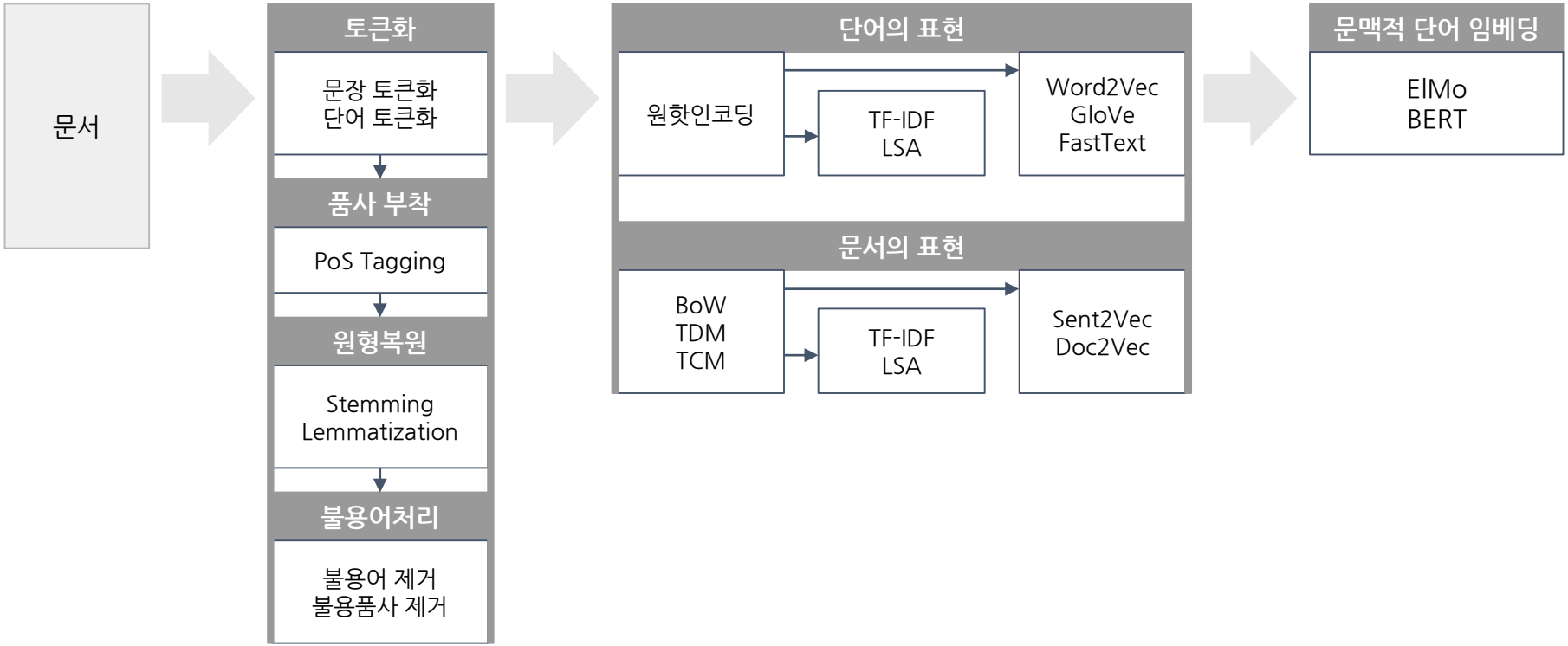


문맥적 단어 임베딩 (Contextualized Word Embedding)

- *좌우 문맥 고려*

좌 → 우	우 → 좌
I play piano.	*I play piano.*
I	*piano*
I → play	*piano→ play*
I → play → piano	*piano → play → I*

임베딩 절차



처리 의존도를 고려하여 성능을 높이는 작업을 진행할 수 있음

BERT 까지

단어의 표현

문맥의 표현

CNN

- 이미지 분류에 많이 사용
- 시계열에 대한 고려 X

RNN

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

LSTM / GRU

- RNN의 gradient vanishing/exploding 문제 해결
- 장기문맥에 대한 고려

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

Tensor

Tensor Ranks, Shapes, and Types

```
t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Rank	Math entity	Python example
0	Scalar (magnitude only)	`s = 483`
1	Vector (magnitude and direction)	`v = [1.1, 2.2, 3.3]`
2	Matrix (table of numbers)	`m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
3	3-Tensor (cube of numbers)	`t = [[[2], [4], [6]], [[8], [10], [12]], [[14], [16], [18]]]`
n	n-Tensor (you get the idea)	`....`

Tensor in NLP

sentence	vector representation
hi John	[[1,0,0,0], [0,1,0,0]]
hi James	[[1,0,0,0], [0,0,1,0]]
hi Brian	[[1,0,0,0], [0,0,0,1]]

(3, 2, 4) 3d tensor!

hi John hi James hi Brian

[[[1,0,0,0], [0,1,0,0]], [[1,0,0,0], [0,0,1,0]], [[1,0,0,0], [0,0,0,1]]]

Tensor in Image

(3, 5, 5)

you have 3 images

5 rows

5 columns