GPT 모델의 이해와 활용

chatGPT와 함께하는 미래 소재 개발의 시작! day 1

최재웅 박사

한국과학기술연구원 계산과학연구센터

2023.08.16

About Me

❖ 최재웅 (Jaewoong Choi)

✓ 산업공학 박사 (2022, SCL@konkuk university, advisor: Janghyeok Yoon)

✓ (Past) Development of machine learning (ML) and natural language processing (NLP)-based system for patent analytics

- Computer Science, applications; Information science & management 분야 SCI 논문 10건 이상 게재, 국내 특허 등록 2건/ 출원 5건, …
- Projects: 기술가치 평가모델 (KIBO), 특허인용추천모델, 특허유지기간 예측모델 (KISTI), 자동이슈 탐지모델 (KISTI), 노이즈 특허 필터링 시스템, 데이터 자동수집 파이프라인 및 데이터베이스 설계 (LX)

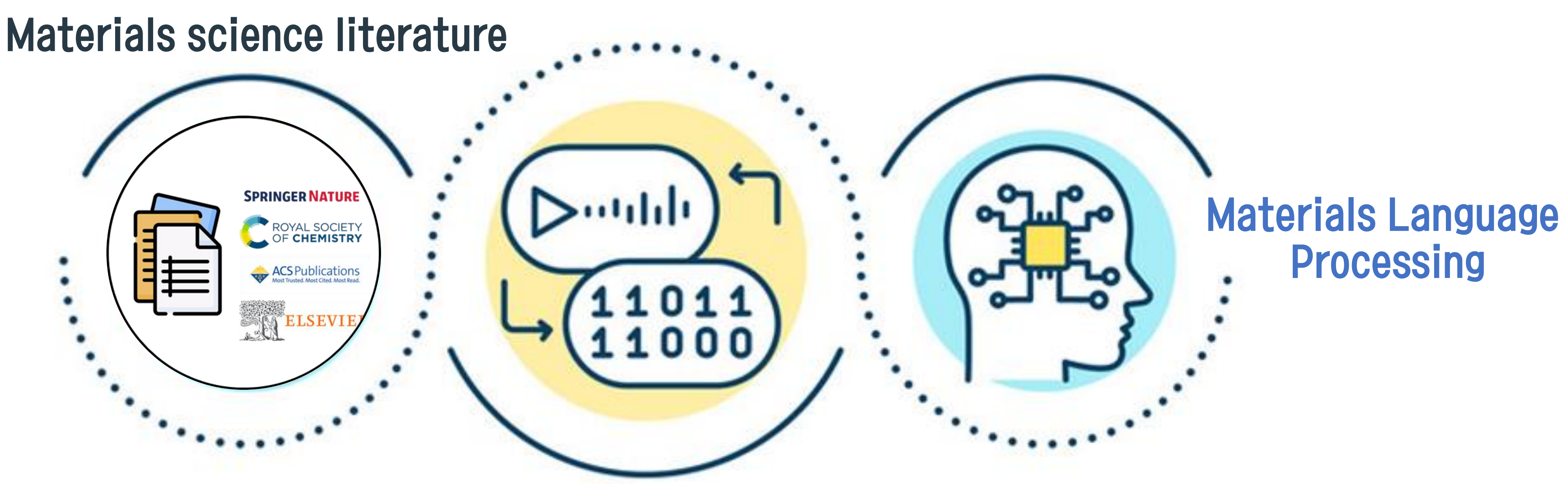
✓ Techniques: data science; machine learning; natural language processing; relational database; …

About Me

❖ 최재웅 (jwchoi95@kist.re.kr)

✓ 박사후 연구원 (2022 ~ 현재), KIST 계산과학연구센터 (advisor: Byungju Lee)

✓ (Now) Applying ML & NLP for materials science literature to extract information such as materials, properties, and synthesis with regard to battery, catalysts, and so on.

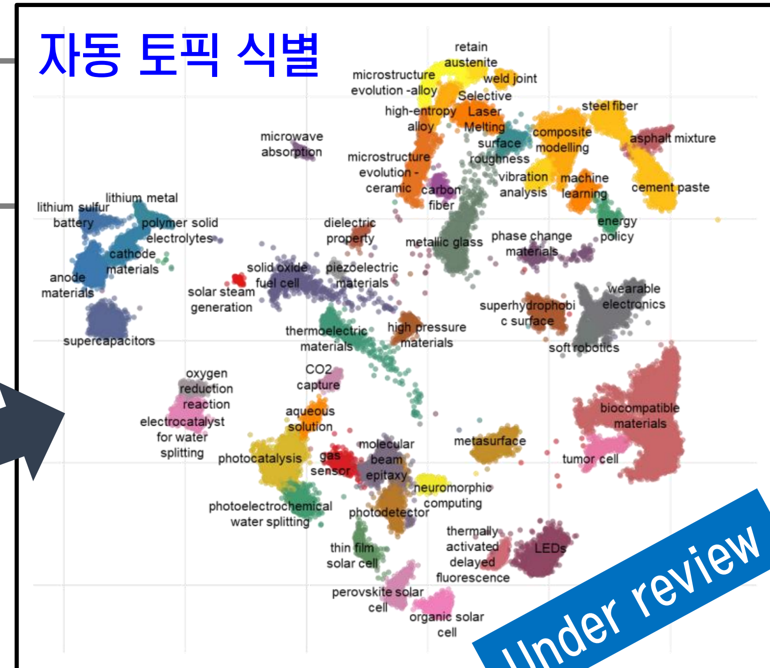


About Me

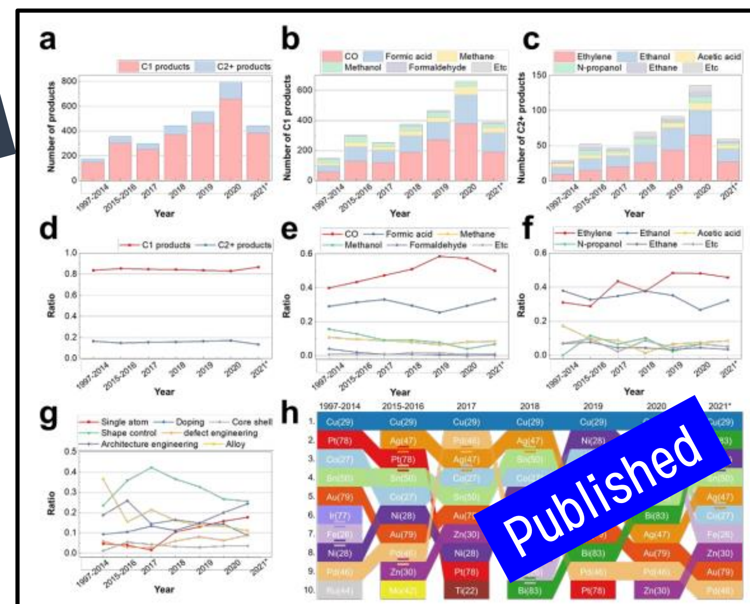
❖ **Current works**

- ✓ Quantitative topic analysis of materials science literature using natural language processing (under review; 1st author)
- ✓ Deep Learning of Electrochemical CO2 Conversion Literature Reveals Research Trends and Directions (published in JMCA (2023); co-author)
- ✓ Accelerated materials language processing enabled by GPT (submitted; 1st author)

Day2 발표 예정



개체명 인식



Today Contents

- ❖ Natural Language Processing (NLP) 소개
- ❖ Language Model (LM) 방법
- ❖ Transformers의 등장
- ❖ BERT & GPT의 비교

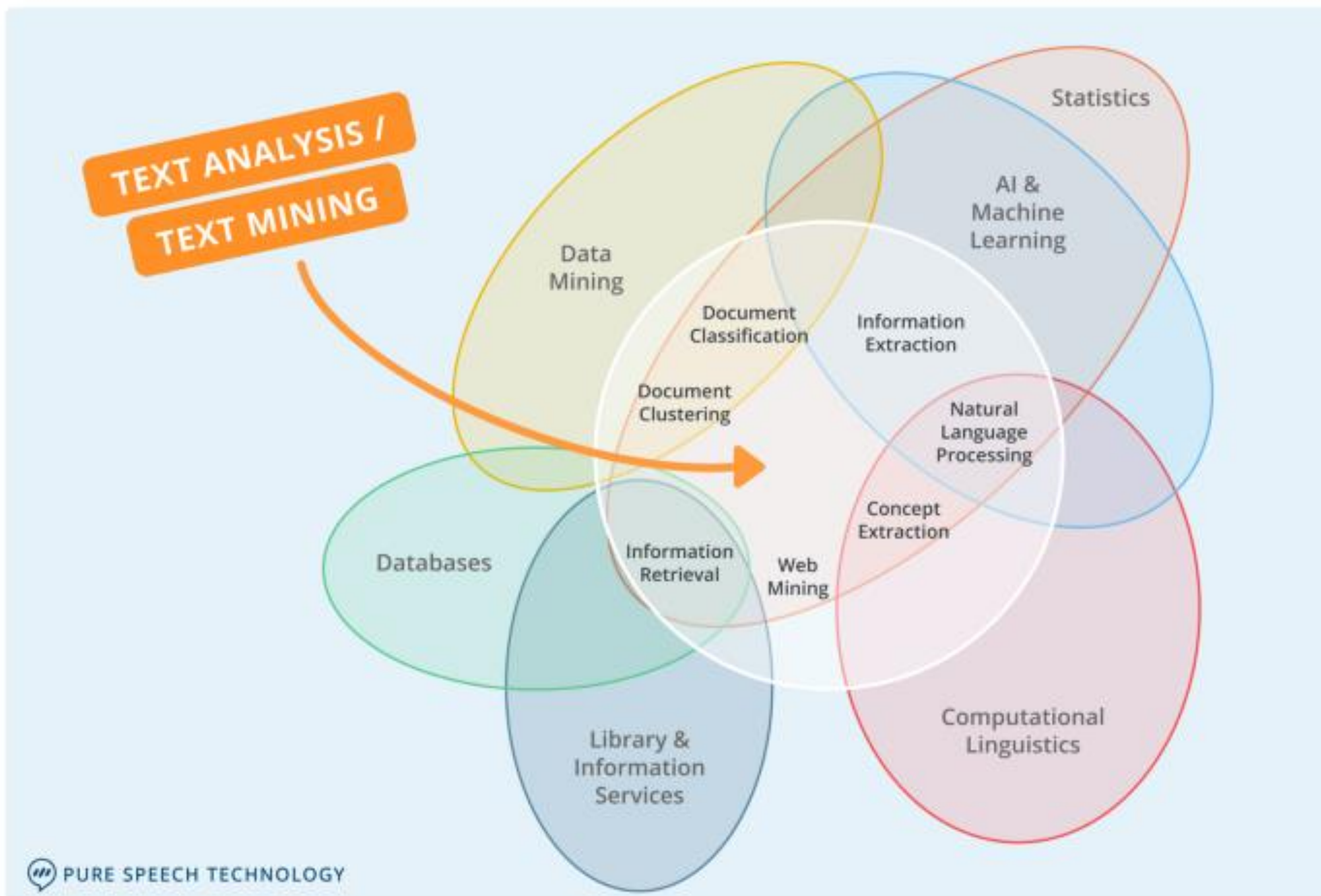
1장 Natural Language processing 소개

GPT 모델의 이해와 활용

chatGPT와 함께하는 미래 소재 개발의 시작 day 1

❖ Natural Language Processing (NLP; 자연어처리)

- ✓ NLP는 언어학, 통계학, 그리고 컴퓨터 사이언스(기계학습, 빅데이터 처리, …)를 포함한다!
- ✓ Layers of
  - Phoneti
  - Morphol
  - Syntax
  - Semanti
  - Pragmat
  - Discour

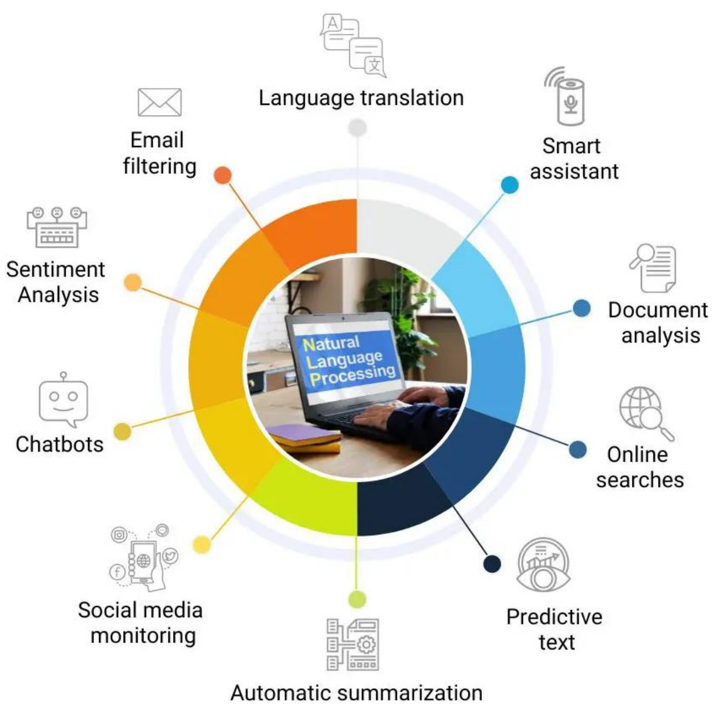


2. Importance of NLP

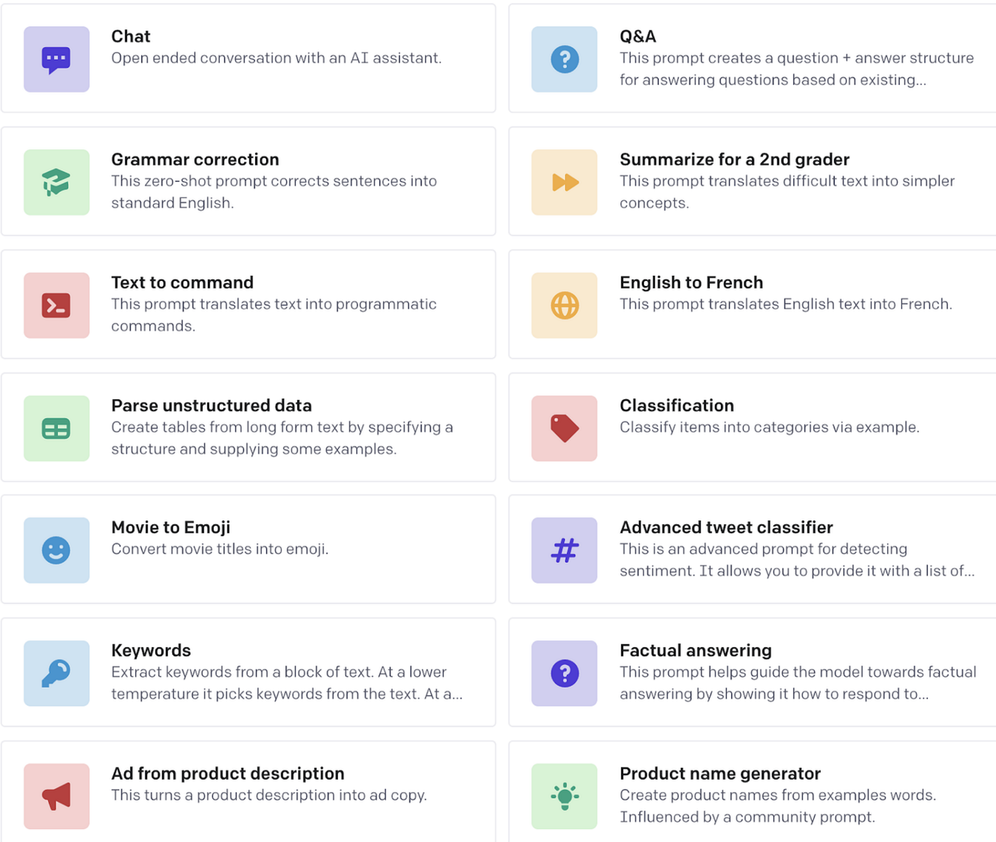
❖ NLP 뭘 할 수 있는가

✓ 기계번역, 텍스트 분류, 감성 분석, 음성 인식, 자동 요약, 질의응답 시스템, 챗봇, 정보 추출, 텍스트 생성 등의 다양한 task에서 높은 성능을 보임

NLP 응용 분야 예시



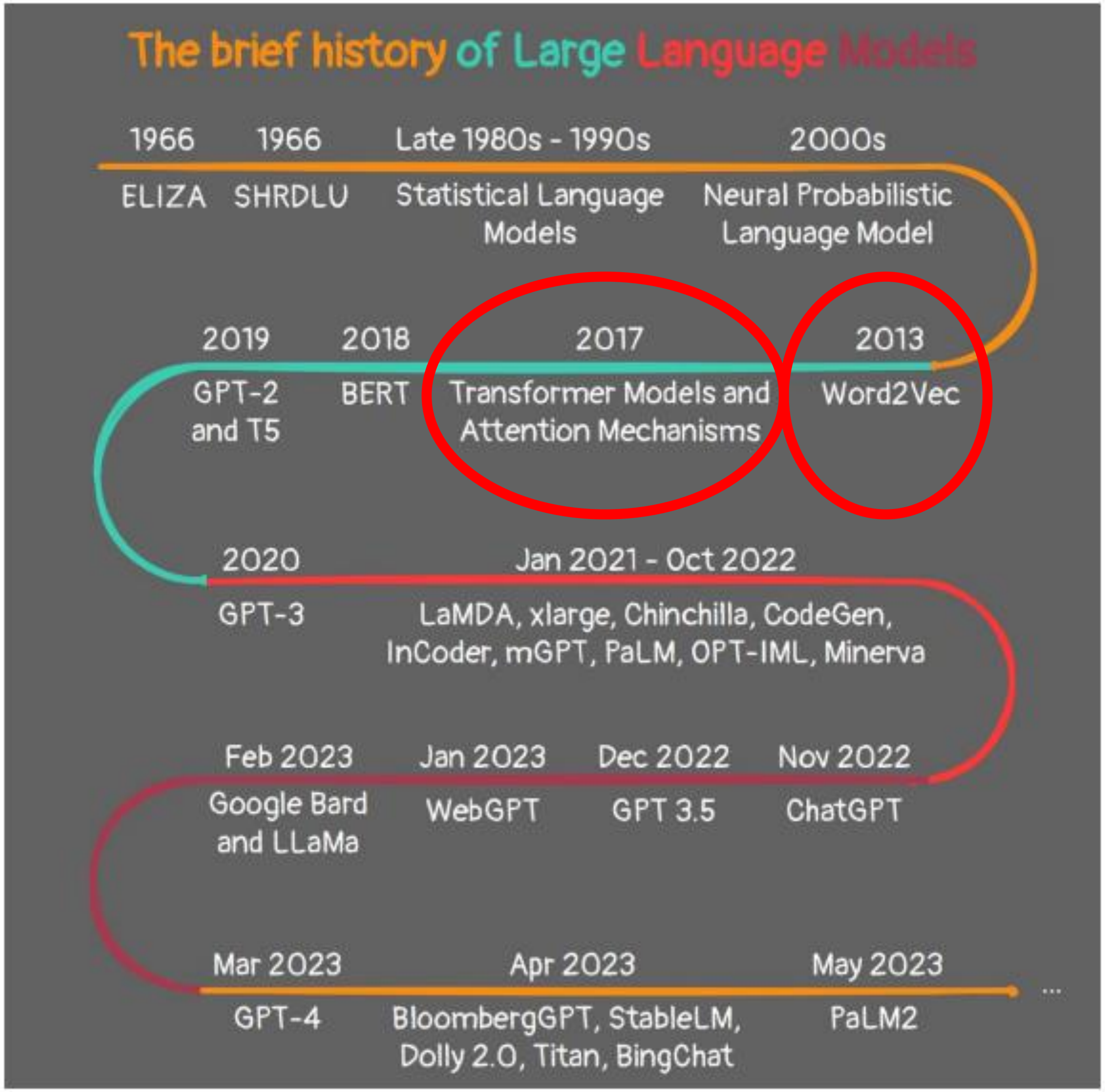
GPT가 제공하는 기능



3. Development of NLP

❖ NLP 기술의 발전

- ✓ 초기: 규칙 기반과 통계적 접근 방식 → 복잡성 • 다양성 해결 X
- ✓ **Deep learning 발전**: Word2vec 과 같이, dense embedding 시도
- ✓ **Transformer 등장**: RNN 계열 모델 한계 해결 + Attention mechanism



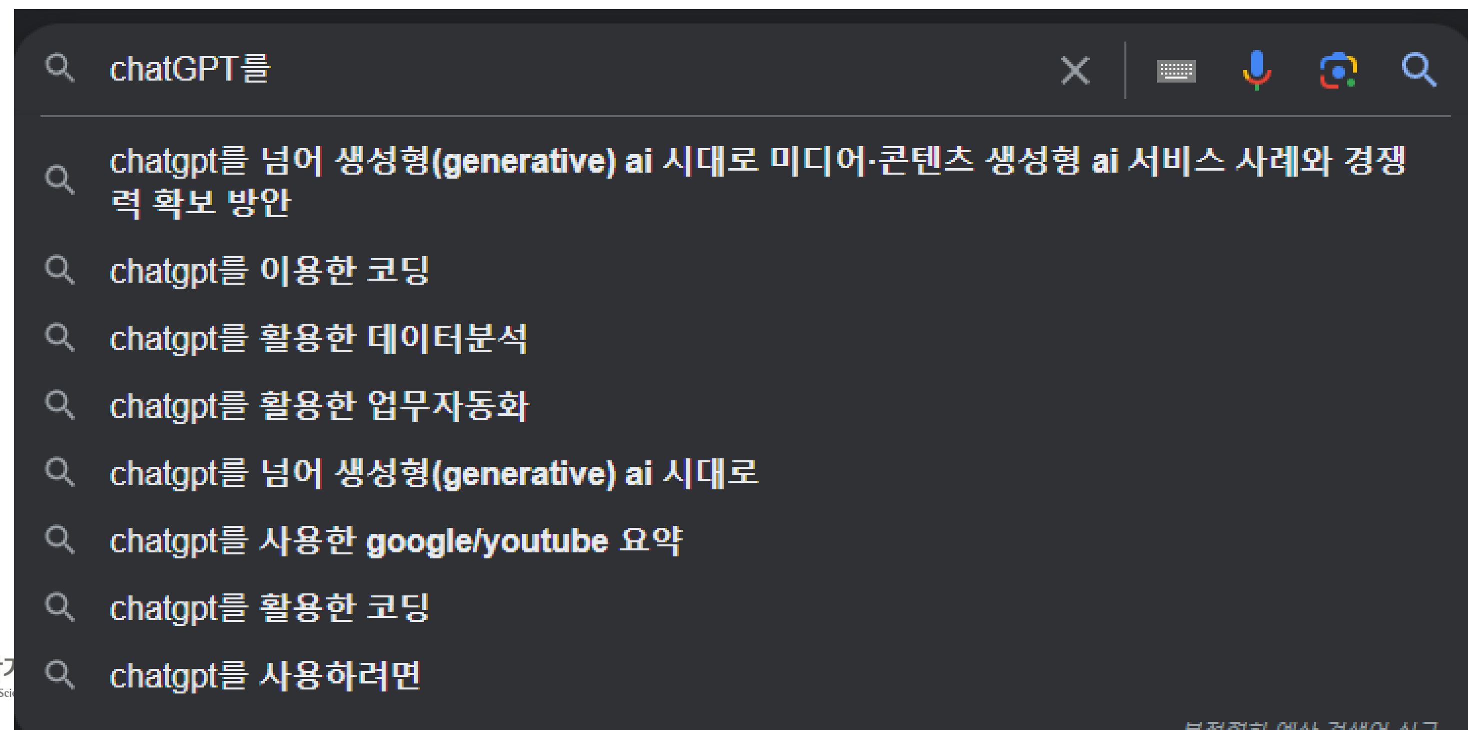
2장 Language model

GPT 모델의 이해와 활용

chatGPT와 함께하는 미래 소재 개발의 시작 day 1

❖ 언어모델은 주어진 단어(토큰)들의 시퀀스(문장 혹은 문서)에 대한 임베딩 (벡터) 표현을 통해, 주제 찾기, 감성 분류, 개체명 인식, 기계 번역, 요약 등의 문제를 풀 수 있음

✓ 아래와 같은 텍스트 생성의 경우, 주어진 문장에서, 다음 단어가 얼마나 자연스러운지 확률적으로 계산하여, 출현하기 적합한 단어를 예측할 수 있음



❖ 통계적 언어모델(Statistical Language Model; SLM)

✓ 자연어를 모델링하기 위해, 단어 시퀀스(문장)에 확률을 할당

- 가장 일반적인 모델링은 이전 단어들을 주고, 다음 단어를 예측

$$P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{n=1}^{n} P(w_n | w_1, \ldots, w_{n-1})$$

P(미래 소재 개발의 시작)

= P(미래) × P(소재|미래) × P(개발의|미래 소재) × P(시작|미래 소재 개발의)

⇒ 희소성 문제 (sparsity problem): 코퍼스에 없으면 풀 수 없다.

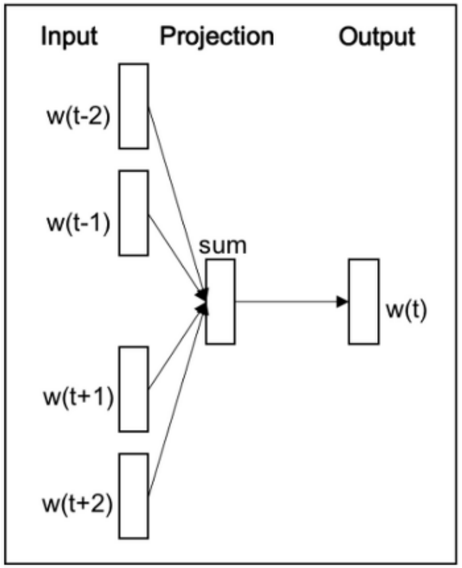
❖ How to Model Language

- ✓ 단어의 임베딩을 목표: Bag of words에서 Neural Network 기반으로
  - Word2vec, FastText ~ Glove
  - CNN 계열:TextCNN, TextCapsulNet, … 주로 텍스트 분류를 목표
- ✓ sequence 데이터 처리 목표: seq2seq 단방향/양방향 ~ 번역 문제!
  - 통계(조건부 확률) 기반 언어모델 (아주 옛날)
  - RNN, LSTM series ~ BiLSTM
  - Transformer 기반: Autoencoding, Autoregressive, Seq2Seq
    - BERT series, GPT series, BART, ELECTRA, …

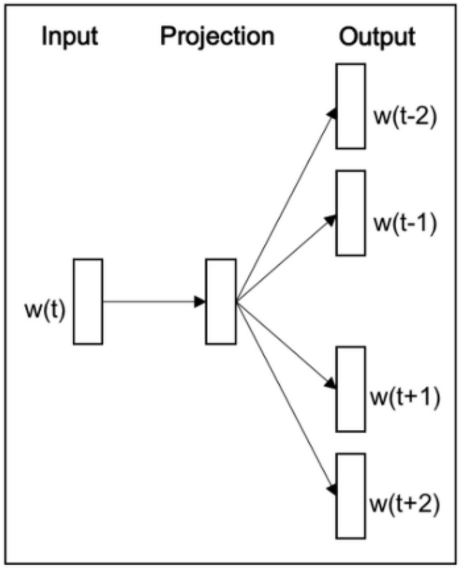
❖ Skip-gram/CBOW Model (Word2Vec)

✓ 중심 단어로 주변단어를 예측하거나(Skip-gram), 주변 단어로 중심단어를 예측(CBOW)하는 Neural Networks 모델 기반

CBOW



Skip-Gram



target word context word

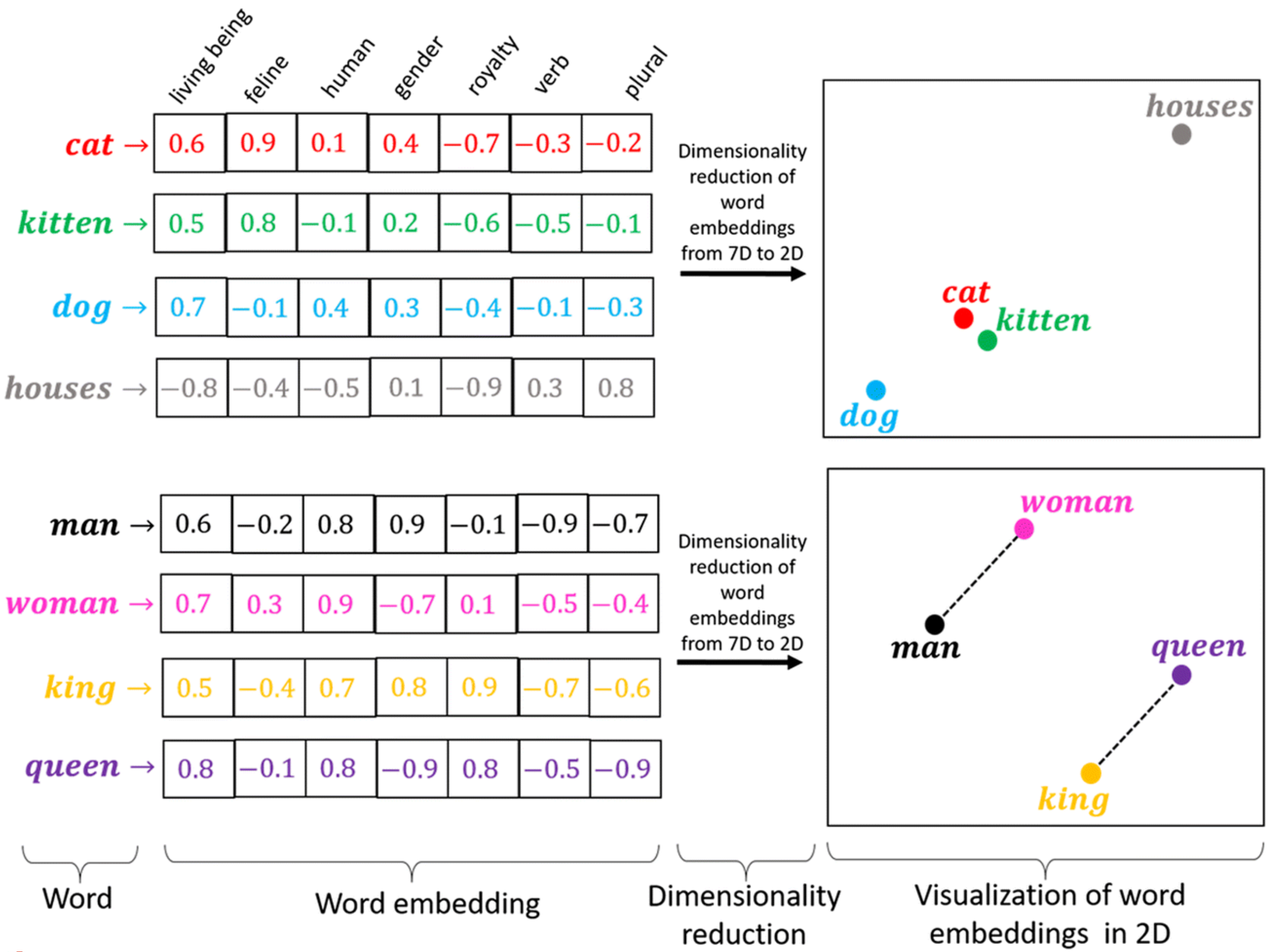
I like natural language processing

I like natural language processing

I like natural language processing

I like natural language processing

❖ Skip-gram/CBOW Model (Word2Vec)

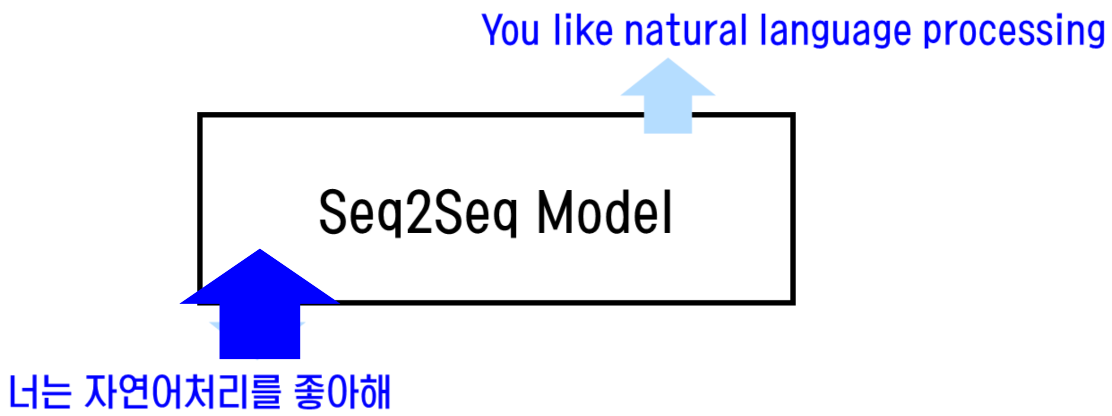


유사한 위치에 있는
단어는 비슷한 의미를
가질 것이다!

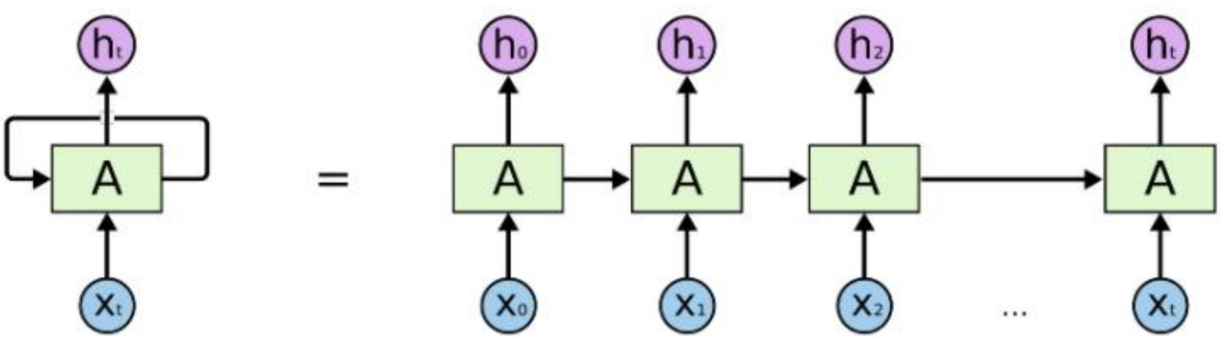
벡터 간
연산(Subtraction)은
단어 간의 관계를
나타낼 수 있음!!

❖ Seq2Seq (단방향/양방향) 언어 모델링

- ✓ 번역 문제처럼, Input sequence를 넣으면 output sequence가 산출되는 형태



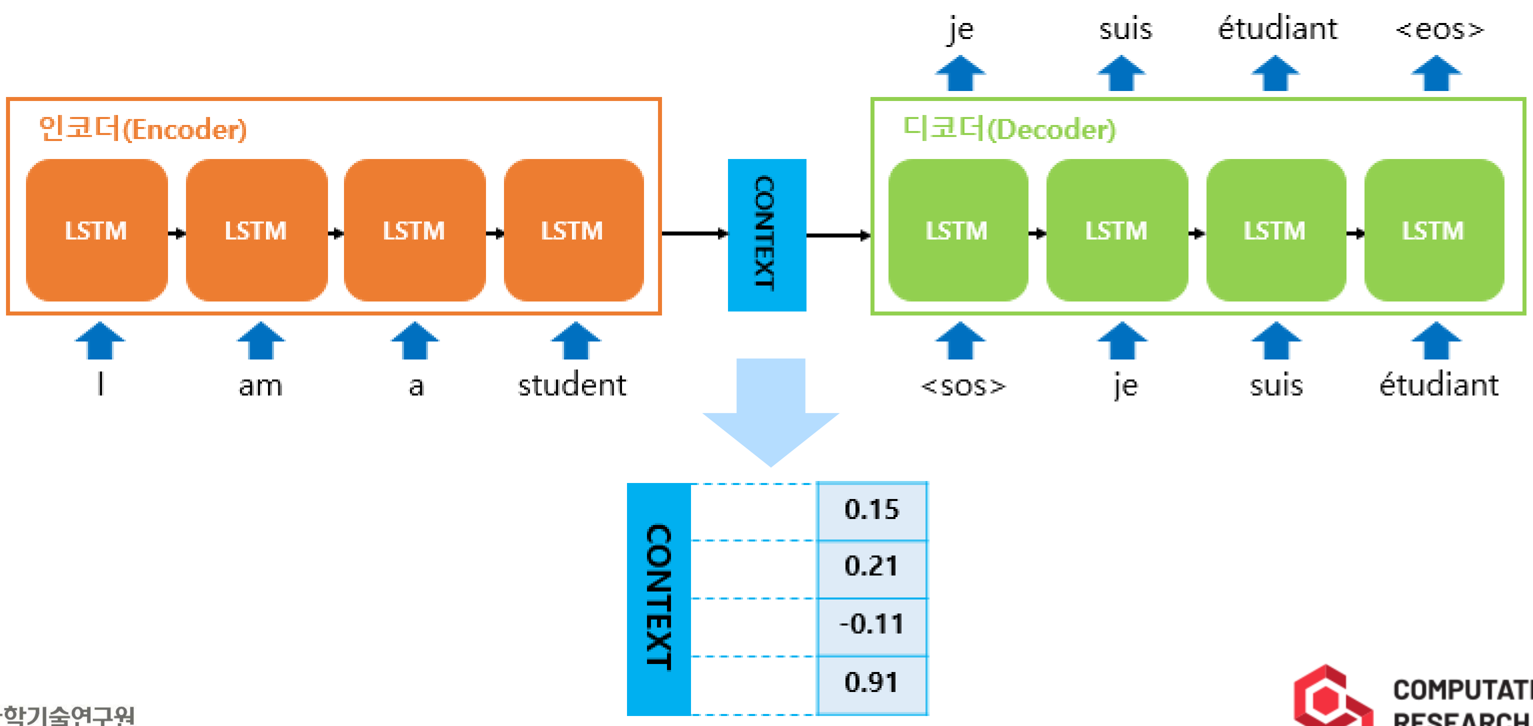
- ✓ Sequential한 데이터 형태를 반영하는 모델 필요! → RNN, LSTM, GRU 등등 사용



❖ Seq2Seq (단방향/양방향) 언어 모델링

✓ Seq2Seq 모델은 크게 encoder와 decoder로 구성

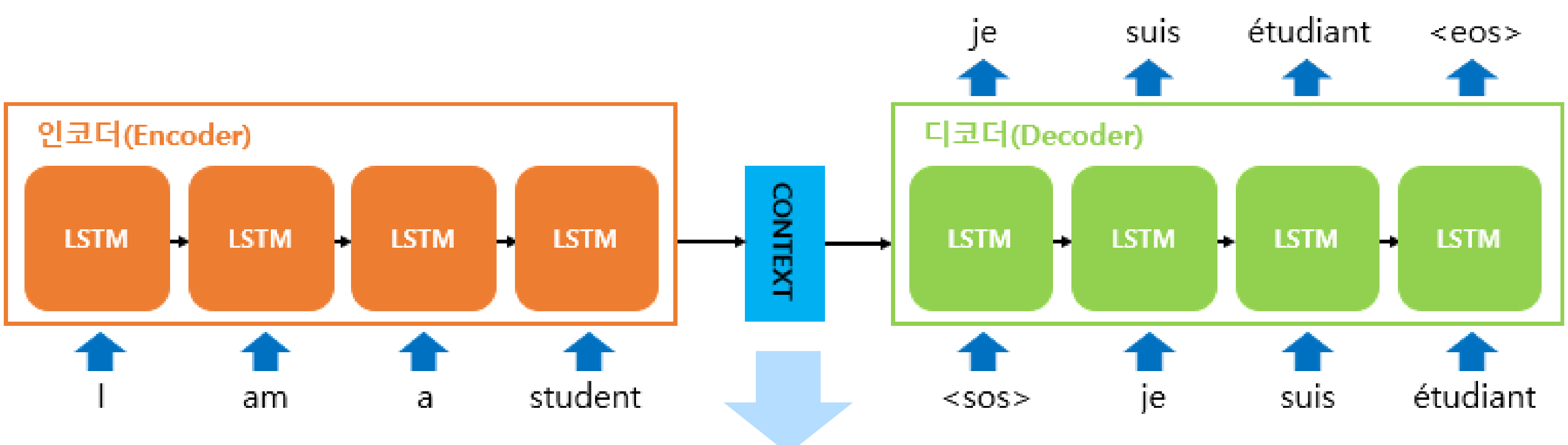
- Encoder는 입력 문장의 모든 단어들을 sequential 하게 입력 받은 뒤에, 압축하여 하나의 Context vector로 만듦
- Decoder는 Context vector로부터 번역된 단어를 하나씩 순차적으로 출력함



Seq2Seq (단방향/양방향) 언어 모델링

✓ RNN, LSTM, GRU 계열 모델 문제점

- 하나의 고정된 크기의 벡터(context vector)에 모든 정보를 압축 ~ 정보 손실
  → seq2seq with attention (2014, 조경현 교수님 bb)
- RNN 모델 구조 특성 상 연속적인 tanh 연산에 따른 Vanishing gradient 문제
  → LSTM, GRU 등의 모델 등장



Advances in neural information processing systems 28 (2015).

입력 시퀀스의 모든 정보를 하나의 벡터로 표현 → Information Loss

3장 Transformer

GPT 모델의 이해와 활용

chatGPT와 함께하는 미래 소재 개발의 시작 day 1

한국과학기술연구원
Korea Institute of Science and Technology

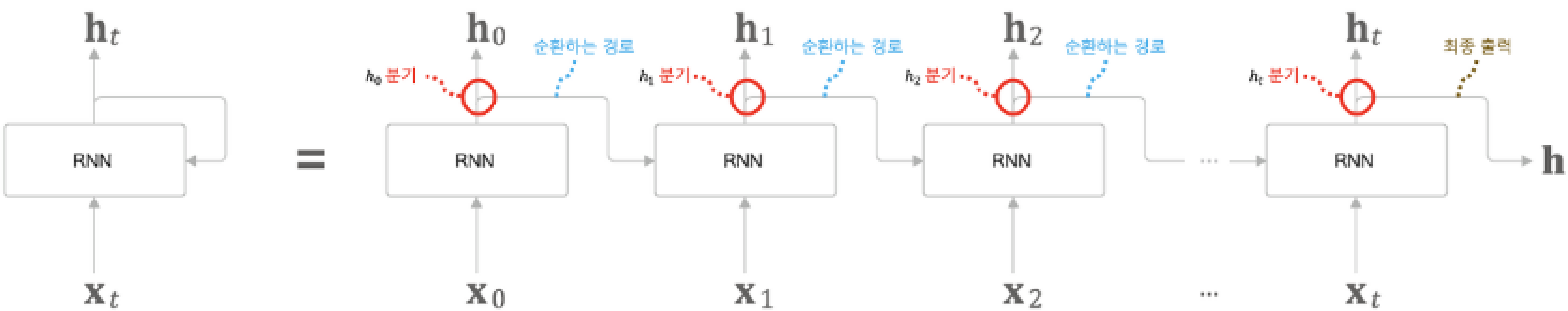
COMPUTATIONAL SCIENCE RESEARCH CENTER
Korea Institute of Science and Technology

❖ RNN 계열의 모델들을 대체하자!

✓ 왜냐면 병렬 처리가 어렵기 때문!

- 이전 state의 계산 결과를 다음 state에서 사용하는 RNN 구조 때문 (Sequential computation)

그림 5-8 RNN 계층의 순환 구조 펼치기



- RNN 계열은 input 시퀀스와 output 시퀀스 간의 단어 대응 관계를 잘 학습하지 못함 (물리적 거리가 먼 경우) ~ Global dependency라고 논문에서 표현

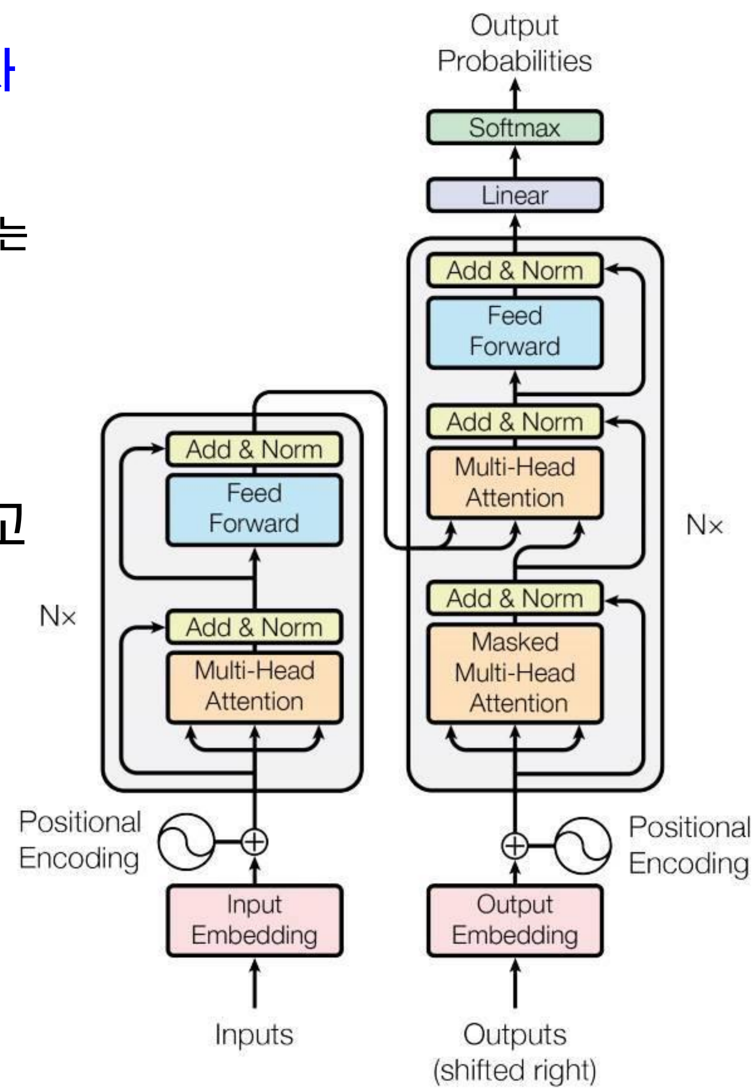
❖ Transformer의 등장

✓ Self Attention mechanism으로 다 바꾸자

- 병렬 처리의 어려움 해결
  - 시퀀스 데이터를 순차적으로 처리할 때 발생하는 계산 복잡도와 연산량 해결
- 입력 시퀀스의 각 단어들이 다른 단어들과 얼마나 관련되어 있는지를 계산

✓ 인코더와 디코더가 텍스트 시퀀스를 이해하고 생성하는 과정

- BERT, GPT 등 대부분의 언어모델의 전신



Attention is All You Need (2017)

❖ Transformer의 등장

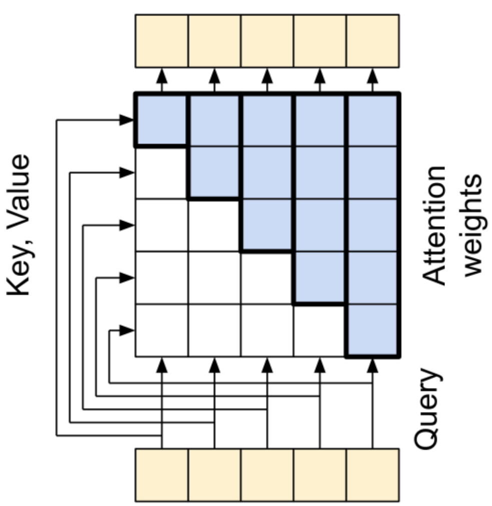
✓ Global dependency를 고려하자

- 기존 seq2seq with attention 모델은 attention을 통해 input과 output을 대응함
- Transformer에서는 self attention을 통해 입력 시퀀스/ 출력 시퀀스 내부 단어 간 대응을 이룸

1. Transformer 모델 개요와 등장배경

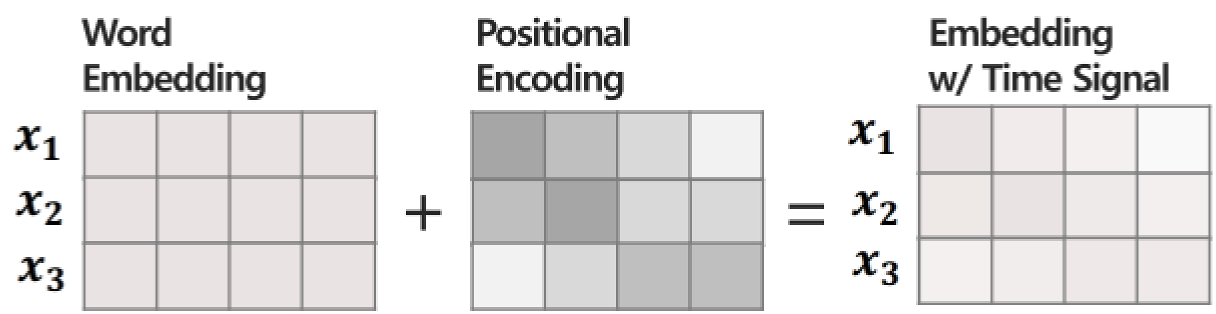
❖ Transformer의 등장

- ✓ RNN 계열은 병렬 처리가 불가능했다
  - 이전 state의 결과를 다음 state에서 input으로 사용하기 때문에, sequence information을 hidden state에 반영하기 위해서 단어별로 계산해야 하기 때문
- ✓ Transformer는 병렬 처리가 가능하다
  - Self-attention으로 입력 시퀀스 내 각 단어 간의 관계를 한번에 계산
  - Position-wise FFNN 형태 (단어별로 독립적으로 적용됨)
  - Position encoding 정보 사용(순차정보 없이 단어 순서 인식)



$$PE_{(pos,i)} = sin\left(\frac{pos}{10000^{2k/d_{model}}}\right) \quad , \ if \ \ i = 2k$$

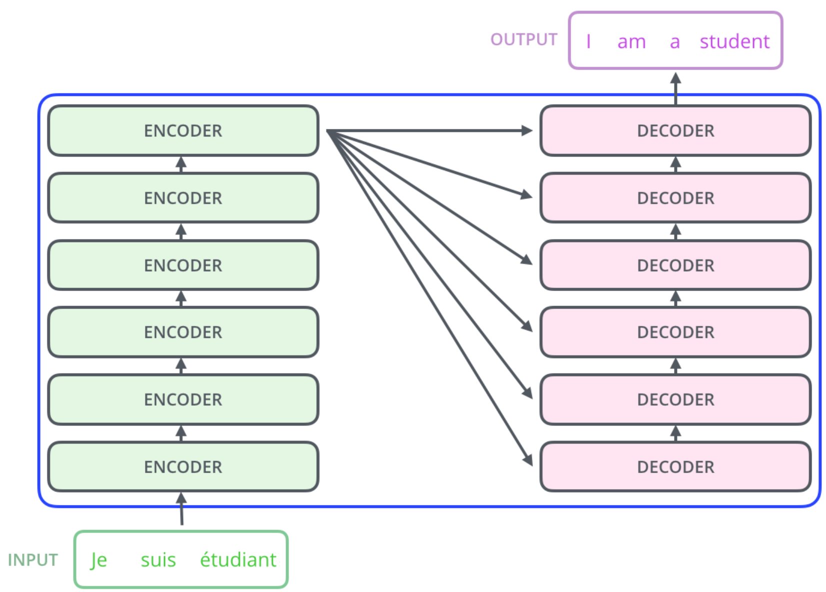
$$PE_{(pos,i)} = cos\left(\frac{pos}{10000^{2k/d_{model}}}\right) \quad , \ if \ \ i = 2k+1$$

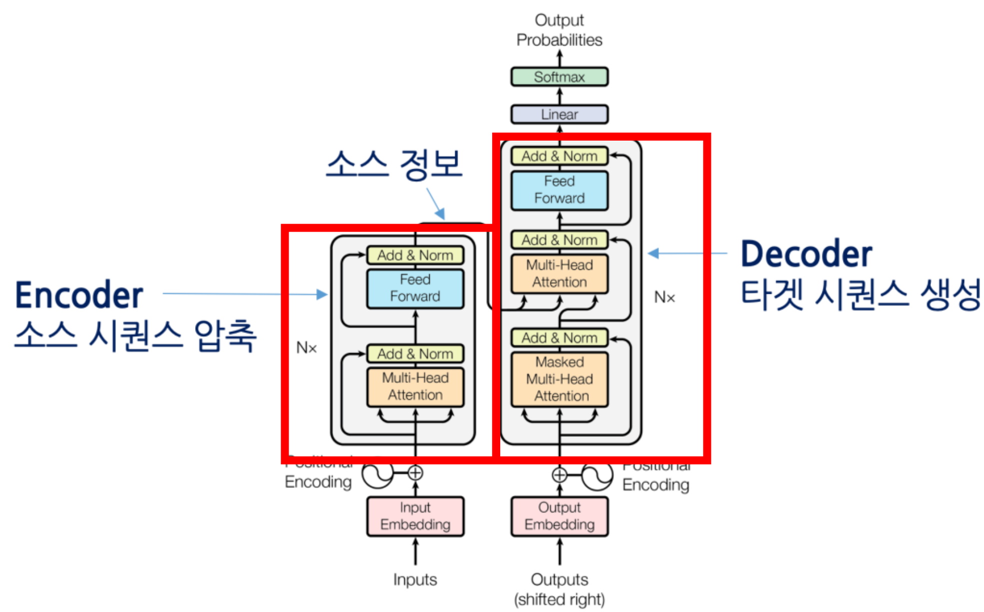


❖ Transformer consists of N modules (Encoder-Decoder)

✓ Seq2Seq 구조 그대로! REMIND

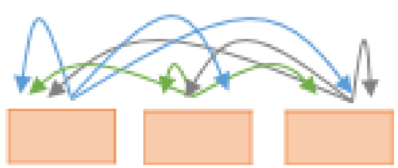
- Machine translation 예시 참고
- Encoder의 역할: input sequence의 정보를 압축!
- Decoder의 역할: output sequence를 생성!



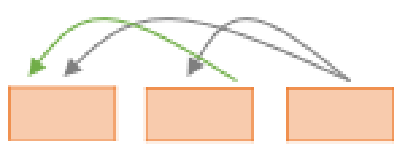


❖ Transformer 내 attention의 역할

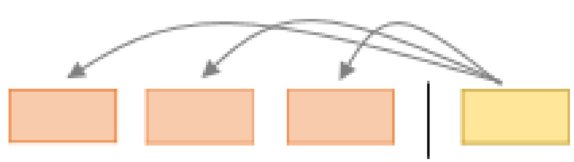
- ✓ Encoder: Multi-head attention ~ BERT 모델로
- ✓ Decoder: Masked Multi-head attention ~ GPT 모델로
- ✓ Encoder-Decoder Attention
  - KEY, VALUE는 encoder에서, QUERY는 decoder에서



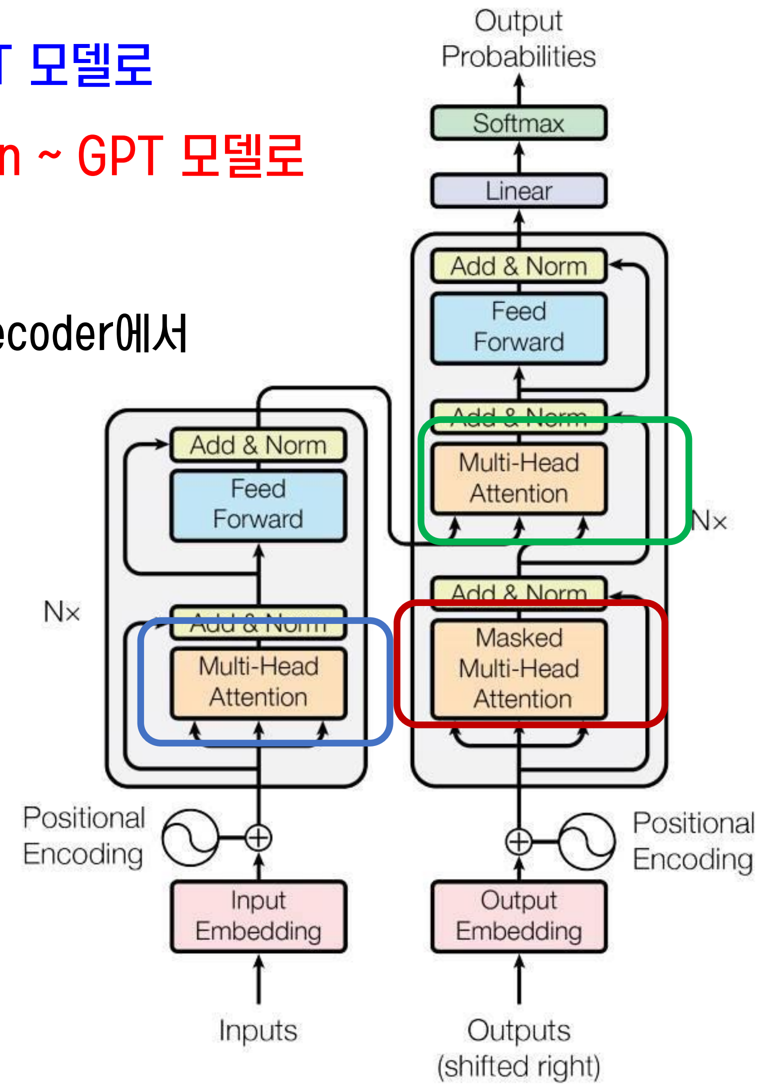
Encoder Self-Attention



Masked Decoder Self-Attention

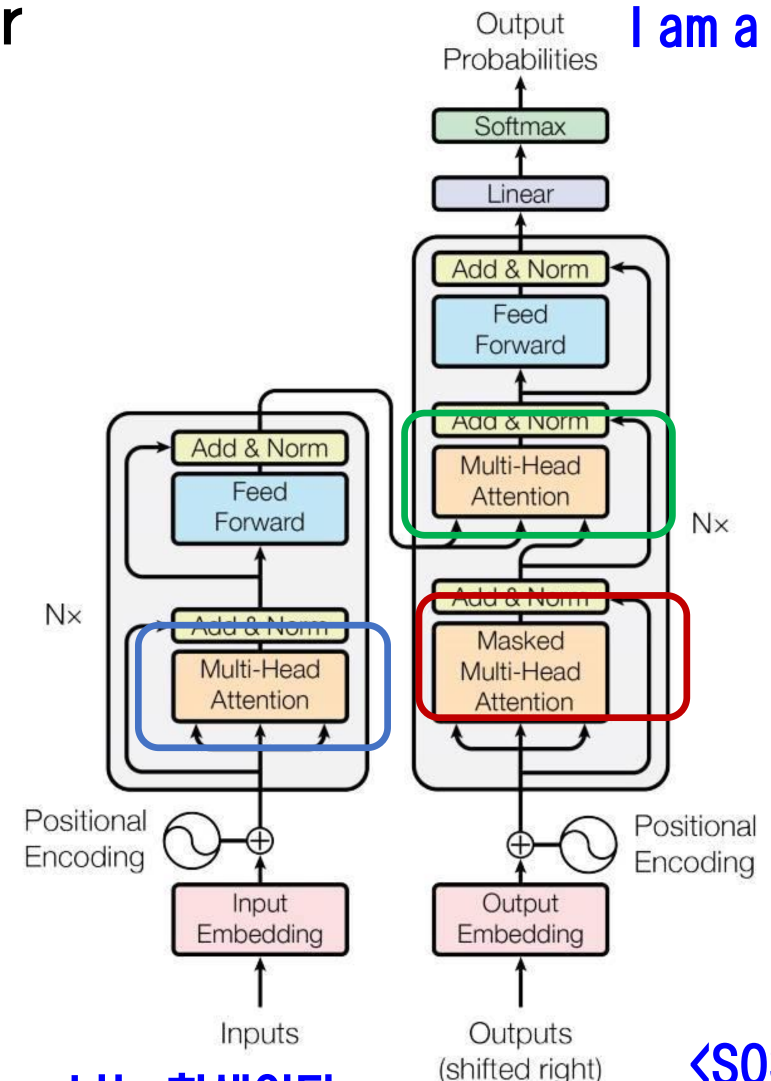


Encoder-Decoder Attention



❖ Transformer

I am a student <EOS>



나는 학생이다

<SOS> I am a student

4장 BERT와 GPT

GPT 모델의 이해와 활용

chatGPT와 함께하는 미래 소재 개발의 시작 day 1

❖ Transformer-based LMs

- ✓ Seq2Seq:
  - Encoder + Decoder
    → T5, BART, Pegasus
- ✓ Autoencoding:
  - Encoder
    → BERT series, ELECTRA
- ✓ Autoregressive:
  - Decoder
    → GPT series, XLNet

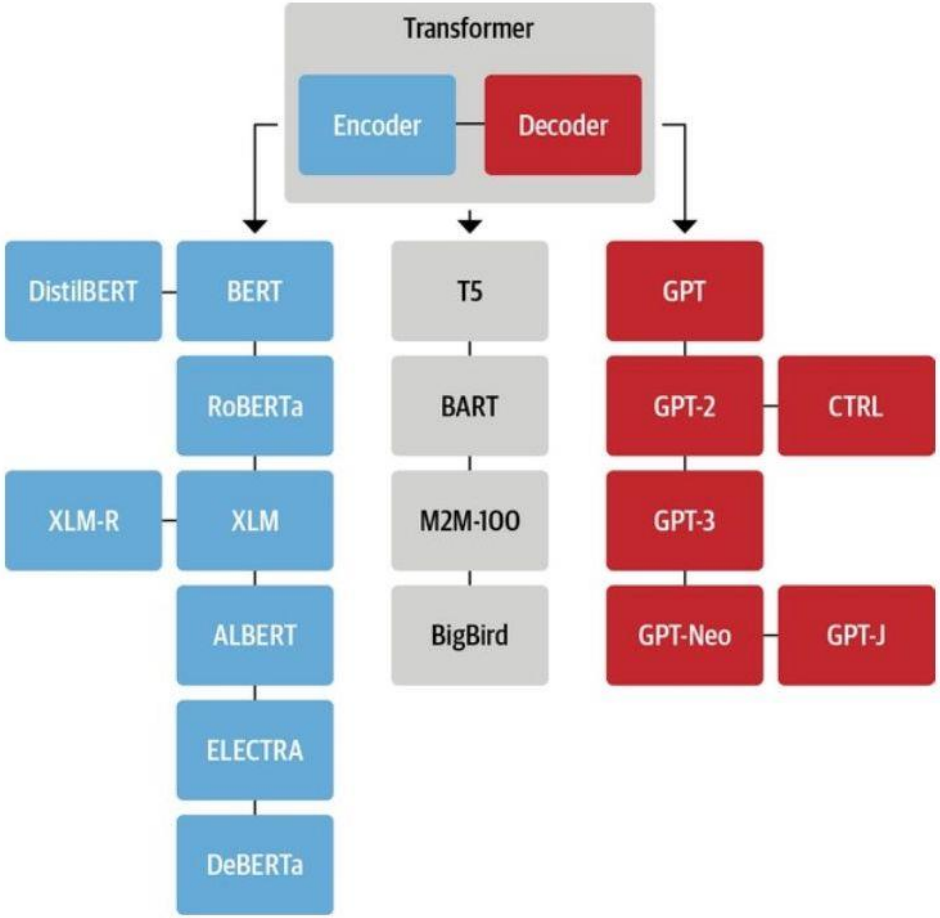
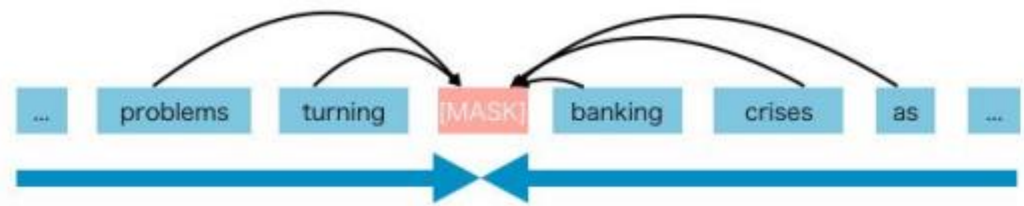


Figure 3-8. An overview of some of the most prominent transformer architectures

❖ Autoencoding Model

- ✓ Trying to reconstruct the original sequence
- ✓ Examples: BERT series model, ELECTRA
- ✓ Natural Bidirectional context, Independent predictions, Artificial Noise



✗ **Fine-tuning discrepancy** caused by [MASK] tokens (not in real data)

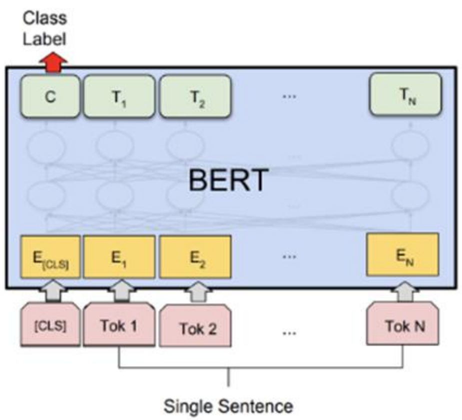
Peter has a [MASK] that does not like [MASK]

Assumes *cat* and *yarn* are independent, which is wrong

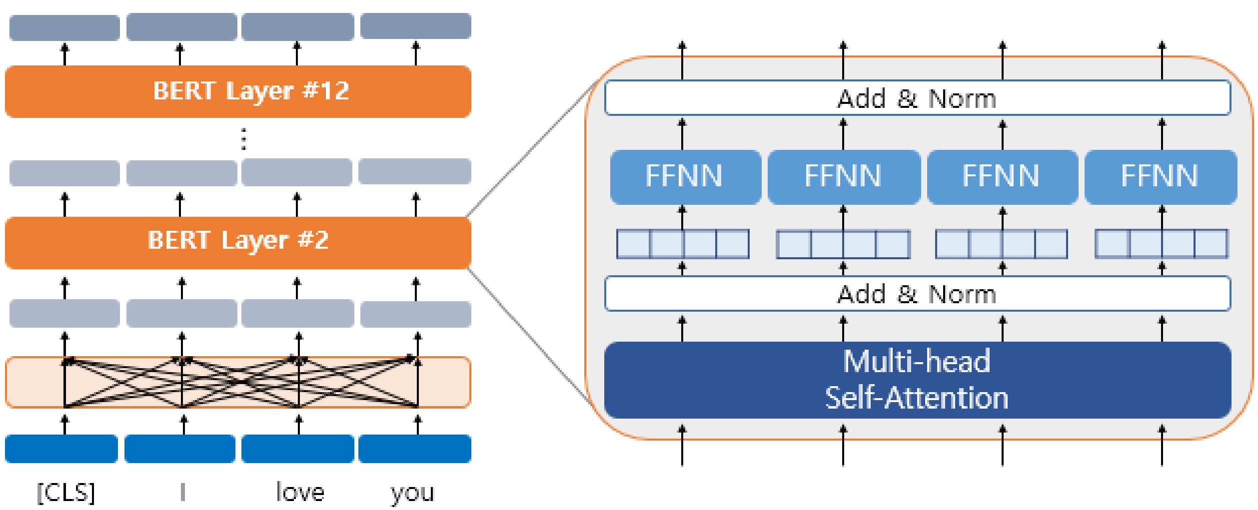
✗ No **joint probability** between masked entries

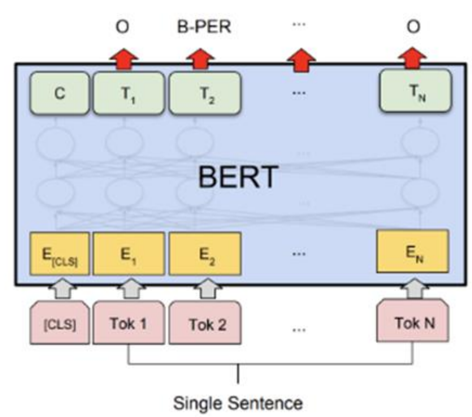
❖ BERT (Bidirectional Encoder Representations from Transformers)

- ✓ BERT uses the encoder of Transformer
  - base model: 12 layers, 768 dimensions, 12 heads
  - large model: 24 layers, 1024 dimensions, 16 heads
- ✓ Task:
  - MLM ~ [MASK] token, NSP ~ [SEP] token
  - Attention mask 1 for real token, 0 for padding token



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





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

❖ 소재문헌에서의 NLP 예시 (BERT in Materials Science)

- ✓ BERT는 어떤 도메인의 데이터셋에 학습했는지에 따라, downstream works에서의 fine-tuning 모델 성능이 달라짐
- ✓ 도메인 맞춤형 BERT 모델의 필요성
  - 특정 도메인에서의 언어 이해(전문 용어, 관계, 내용); 예를 들어, biomedical 분야는 bio 전용 NLP모델 (BioBERT, PubMedBERT 등) 을 통해 의료 진단, 의약품 효능 분석을 수행함

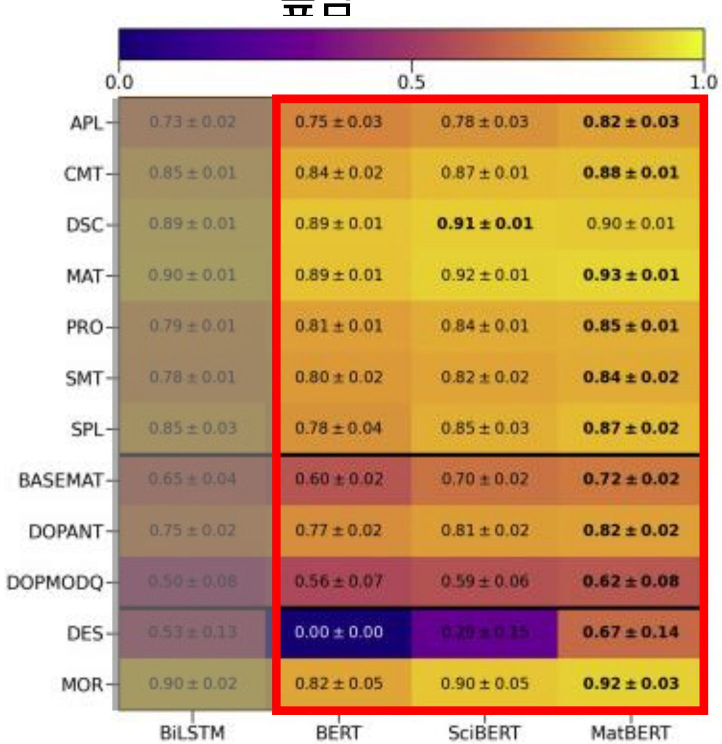
Biomedical Term	Category	BERT	SciBERT	PubMedBERT (Ours)
diabetes	disease	✓	✓	✓
leukemia	disease	✓	✓	✓
lithium	drug	✓	✓	✓
insulin	drug	✓	✓	✓
DNA	gene	✓	✓	✓
promoter	gene	✓	✓	✓
hypertension	disease	hyper-tension	✓	✓
nephropathy	disease	ne-ph-rop-athy	✓	✓
lymphoma	disease	l-ym-ph-oma	✓	✓
lidocaine	drug	lid-oca-ine]	✓	✓
oropharyngeal	organ	oro-pha-ryn-ge-al	or-opharyngeal	✓
cardiomyocyte	cell	card-iom-yo-cy-te	cardiomy-ocyte	✓
chloramphenicol	drug	ch-lor-amp-hen-ico-l	chlor-amp-hen-icol	✓
RecA	gene	Rec-A	Rec-A	✓
acetyltransferase	gene	ace-ty-lt-ran-sf-eras-e	acetyl-transferase	✓
clonidine	drug	cl-oni-dine	clon-idine	✓
naloxone	drug	na-lo-xon-e	nal-oxo-ne	✓

유의미한 tokenizer 차이
+
유의미한 downstream works
성능 차이

❖ 소재문헌에서의 NLP 예시 (BERT in Materials Science)

✓ 사례: 소재문헌에 맞춤화된 BERT 모델

- 도메인 맞춤형 BERT 모델의 필요성 → MatBERT, MaterialsBERT, MatSciBERT, …
- Tokenizer: Byte Pair Encoding 알고리즘에 따라 전체 데이터 셋 내에 문자열의 상대적 빈도수에 기반하여, 텍스트를 하위 단위로 분할하여 희소성을 줄이고, 효율성을 높임



원본 텍스트: "The electrolyte (Merck) was 1M LiPF6 in a 1:1 (weight ratio) ethylene carbonate: di-methyl carbonate (EC: DMC) mixture."

BERT (일반적인 텍스트(Wikipedia, news 등)로 학습한 모델):

['The', 'electro', '##ly', '##te', '(' , 'Me', '##rc', '##k', ')', 'was', '1', 'M', 'Li', '##P', '##F', …]

MatBERT (소재문헌으로 학습한 모델):

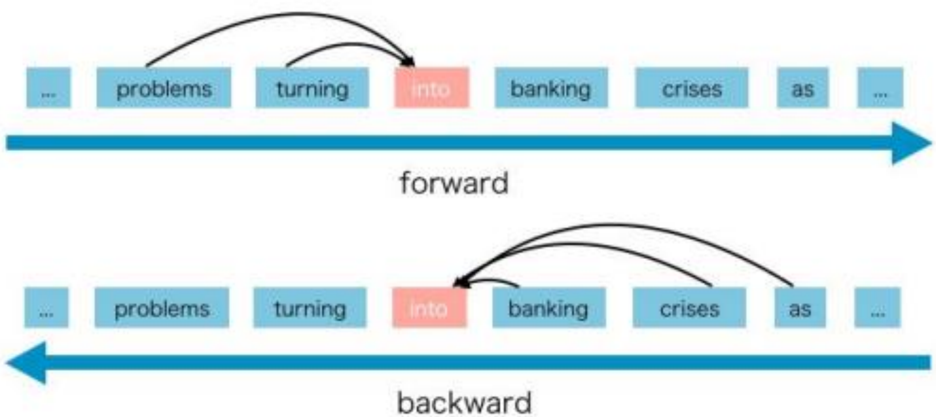
['The', 'electrolyte', '(', 'Merck', ')', 'was', '1', 'M', 'LiFP6', …]

COMPUTATIONAL SCIENCE RESEARCH CENTER
Korea Institute of Science and Technology

❖ Autoregressive model (Causal LM)

- ✓ Using the context word to predict the next word
  - by estimating the probability distribution of a text corpus
- ✓ Examples: XLNet, GPT series

Use context to predict the next word



$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid \mathbf{x}_{<t})$$

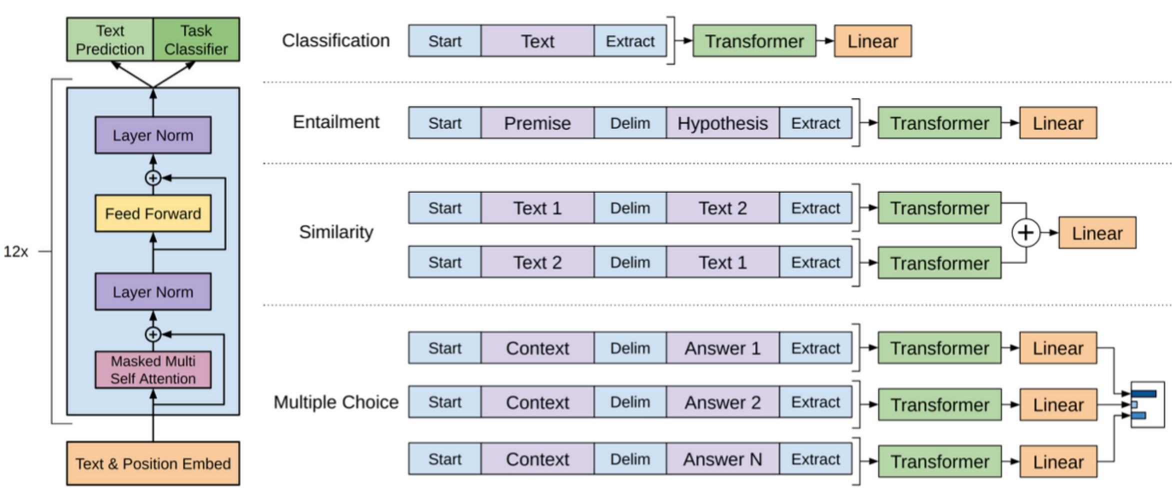
$$p(\mathbf{x}) = \prod_{t=T}^{1} p(x_t \mid \mathbf{x}_{>t})$$

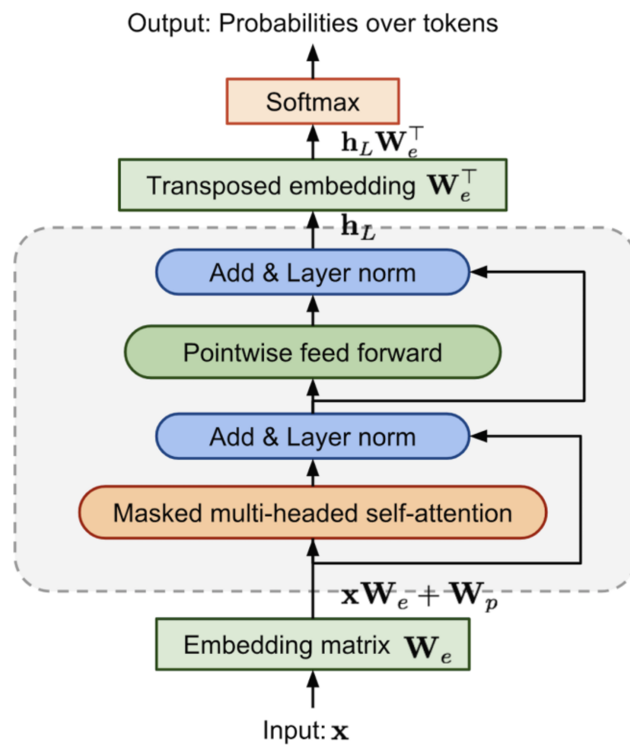
Only considers context in one direction

2. GPT 모델

❖ Autoregressive model

- ✓ GPT is N-stacked decoders of Transformer
  - (encoder-decoder attention) also removed
- ✓ GPT consists of unsupervised pre-training and supervised fine-tuning
  - 12 decoder blocks

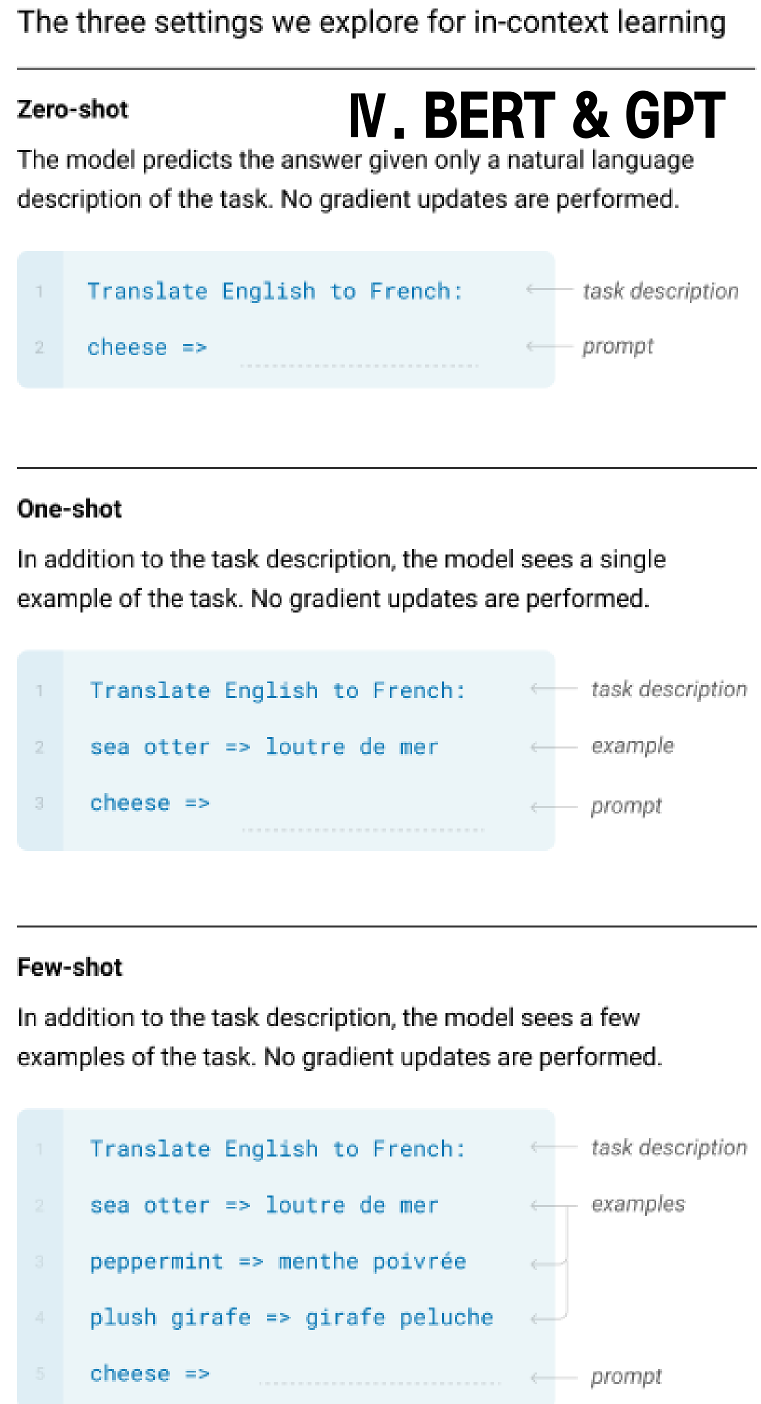




3. BERT와 GPT 어떻게 쓸까

❖ GPT의 특성

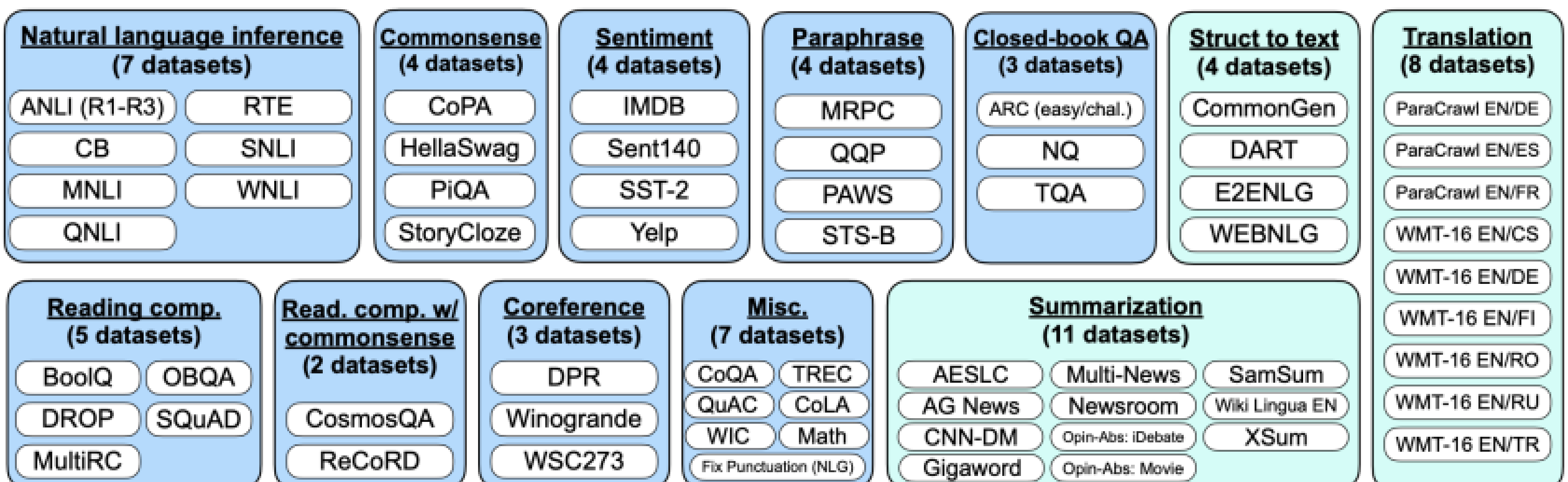
- ✓ Few shot learner로써의 역할
  - 소수의 N 개 혹은 0개로 학습
  - N-shot K-way learning
- ✓ Task description, Example, Input 으로 구성
- ✓ In-context learning
  - 전체 Prompt의 내용으로, 주어진 Context를 이해하고, 답변을 생성
    → weight update X
  - 즉, 주어진 텍스트 내 패턴을 학습해 결과를 generate하자!



❖ 어떤 모델을 사용해야 하는가?

✓ Closed-domain task에서는 여전히 BERT 기반의 supervised model들이 여전히 강세

- Closed-domain task는 question answering, sentiment analysis, machine translation, summarization, information extraction 등 전통적인 task로, 공개 데이터셋 존재



❖ 어떤 모델을 사용해야 하는가?

✓ Open-domain task에서는 생성형 모델들이 잘 해결하는 문제들이 존재

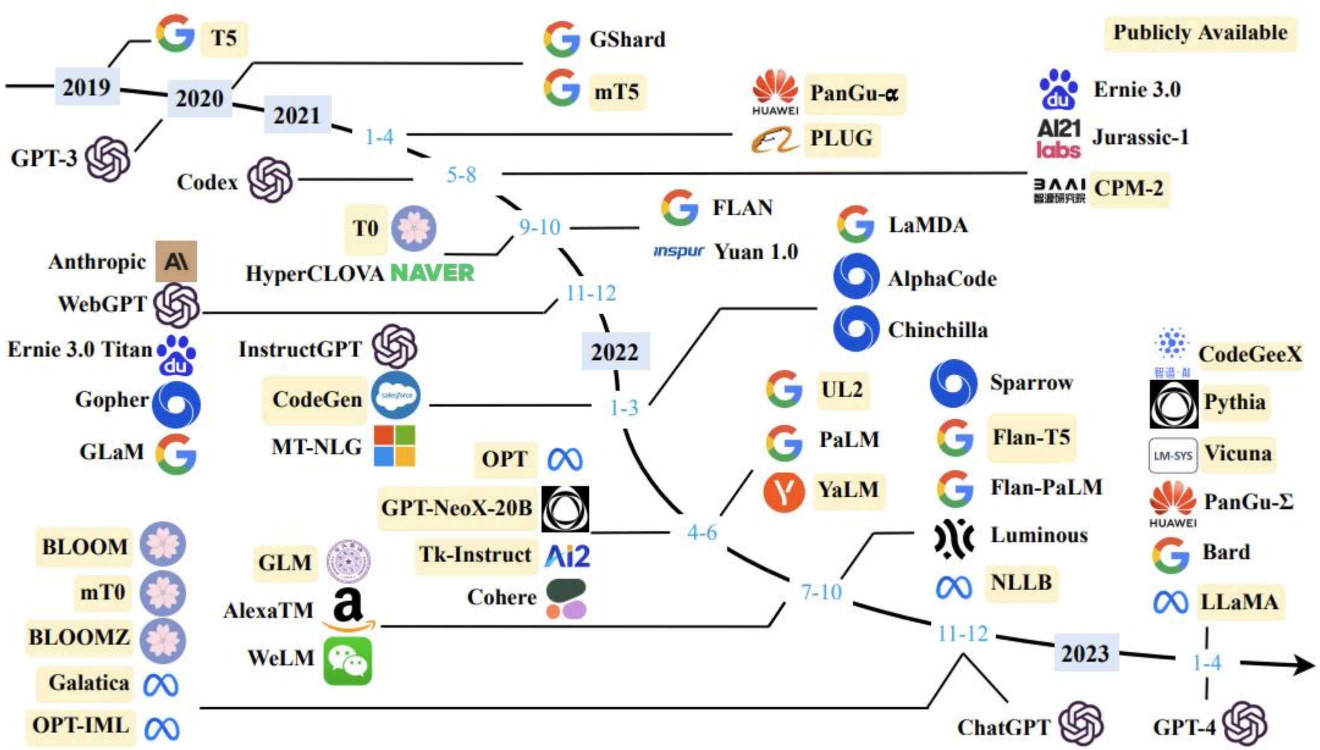
- Open-domain task는 simple arithmetic, fact-based question, common sense reasoning, historical facts, analogy making, visual reasoning 등 잠재적 답변이 존재하는 상황

Dataset	Example	Article / Paragraph
SQuAD	**Q**: How many provinces did the Ottoman empire contain in the 17th century? **A**: 32	**Article**: Ottoman Empire **Paragraph**: ... At the beginning of the 17th century the empire contained 32 provinces and numerous vassal states. Some of these were later absorbed into the Ottoman Empire, while others were granted various types of autonomy during the course of centuries.
CuratedTREC	**Q**: What U.S. state's motto is "Live free or Die"? **A**: New Hampshire	**Article**: Live Free or Die **Paragraph**: "Live Free or Die" is the official motto of the U.S. state of New Hampshire, adopted by the state in 1945. It is possibly the best-known of all state mottos, partly because it conveys an assertive independence historically found in American political philosophy and partly because of its contrast to the milder sentiments found in other state mottos.
WebQuestions	**Q**: What part of the atom did Chadwick discover?[†] **A**: neutron	**Article**: Atom **Paragraph**: ... The atomic mass of these isotopes varied by integer amounts, called the whole number rule. The explanation for these different isotopes awaited the discovery of the neutron, an uncharged particle with a mass similar to the proton, by the physicist James Chadwick in 1932. ...
WikiMovies	**Q**: Who wrote the film Gigli? **A**: Martin Brest	**Article**: Gigli **Paragraph**: Gigli is a 2003 American romantic comedy film written and directed by Martin Brest and starring Ben Affleck, Jennifer Lopez, Justin Bartha, Al Pacino, Christopher Walken, and Lainie Kazan.

❖ Large Language Model (LLM)

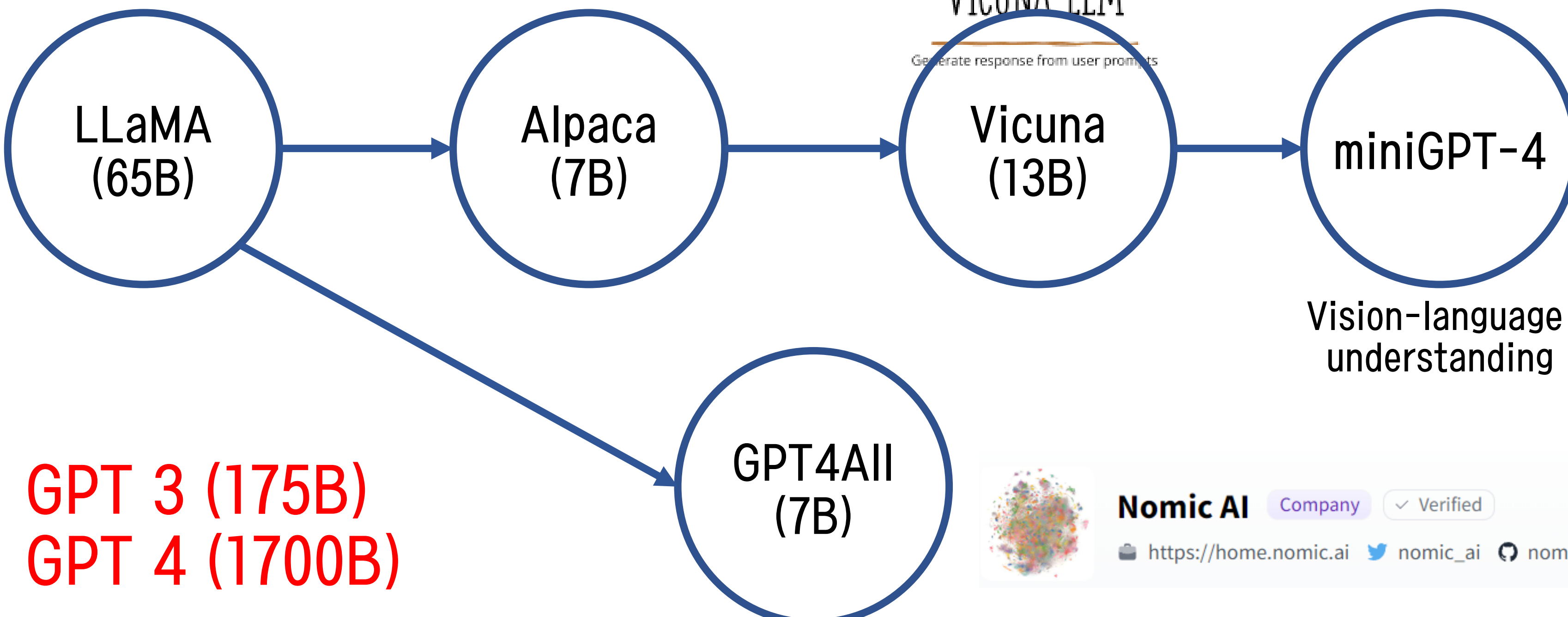
✓ 수십억개 이상의 문장, 웹페이지, 뉴스기사 등의 텍스트 학습, 즉, 사전 훈련(pre-training)을 진행한 언어모델

- 이후에, 특정 태스크에 맞게 미세 조정(fine-tuning)하여 사용



❖ LLM 앞으로 어떻게 될 것인가 (낙타 농장의 서막,,,)

✓ 폐쇄형 모델(Closed & Heavy) → 개방형 모델(Open Source & Light)



GPT 3 (175B)
GPT 4 (1700B)

LLM 모델 요약

- Autoencoding Model의 인코딩 정보로, supervised model을 개발하는 것이 여전히 SOTA이고, 필요하다

- Autoregressive Model의 생성 능력으로 open domain task를 푸는 것은 가능하다

- GPT의 in-context learning으로, 복잡한 NLP task를 풀기 위한 전략적인 prompt 설계가 필요하다

감사합니다.
Q&A

최재웅 박사

(jwchoi95@kist.re.kr)

한국과학기술연구원 계산과학연구센터

Papers, websites, …

- P4: Deep Learning of Electrochemical CO 2 Conversion Literature Reveals Research Trends and Directions. Journal of Materials Chemistry A (2023)
- P7: https://www.purespeechtechnology.com/text-analysis-text-analytics-text-mining/
- P8: https://openai.com/
- P8:Applications of Natural Language Processing | Data Science Dojo
- P9: https://levelup.gitconnected.com/the-brief-history-of-large-language-models-a-journey-from-eliza-to-gpt-4-and-google-bard-167c614af5af
- P13: https://wikidocs.net/21687
- P15: DL] Word2Vec, CBOW, Skip-Gram, Negative Sampling
- P16: Training Word2vec using gensim
- P17: Tshitoyan, Vahe, et al. "Unsupervised word embeddings capture latent knowledge from materials science literature." Nature 571.7763 (2019): 95-98.
- P19: Day 01 Basics of Sequential Modelling , NLP and Large Language Models(LLM)
- P20-21: Cortes, Corinna, et al. "Advances in neural information processing systems 28." Proceedings of the 29th Annual Conference on Neural Information Processing Systems. 2015.
- P22: https://velog.io/@sujeongim/NLPSeq2Seq-with-Attention
- P23: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
- P24, P28 : https://techblog-history-younghunjo1.tistory.com/481

- P25: https://huidea.tistory.com/237
- P29, P33-35: Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- P30: https://techblog-history-younghunjo1.tistory.com/496
- P31: Neural machine translation with a Transformer and Keras | Text | TensorFlow
- P32: https://ratsgo.github.io/nlpbook/docs/language_model/transformers/
- P33: 16-01 트랜스포머(Transformer) - 딥 러닝을 이용한 자연어 처리 입문
- P40: Aman's AI Journal • Primers • Autoregressive vs. Autoencoder Models
- P41, P47: Aman's AI Journal • Primers • Autoregressive vs. Autoencoder Models
- P42: RUBERT: A Bilingual Roman Urdu BERT Using Cross Lingual Transfer Learning
- P43: ACM Transactions on Computing for Healthcare (HEALTH) 3, no. 1 (2021): 1-23.
- P44: Trewartha, Amalie, et al. "Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science." Patterns 3.4 (2022).
- P47: https://paperswithcode.com/method/gpt
- P48: Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
- P51-52: Wei, Jason, et al. "Finetuned language models are zero-shot learners." arXiv preprint arXiv:2109.01652 (2021).
- P53: https://github.com/hollobit/GenAI_LLM_timeline