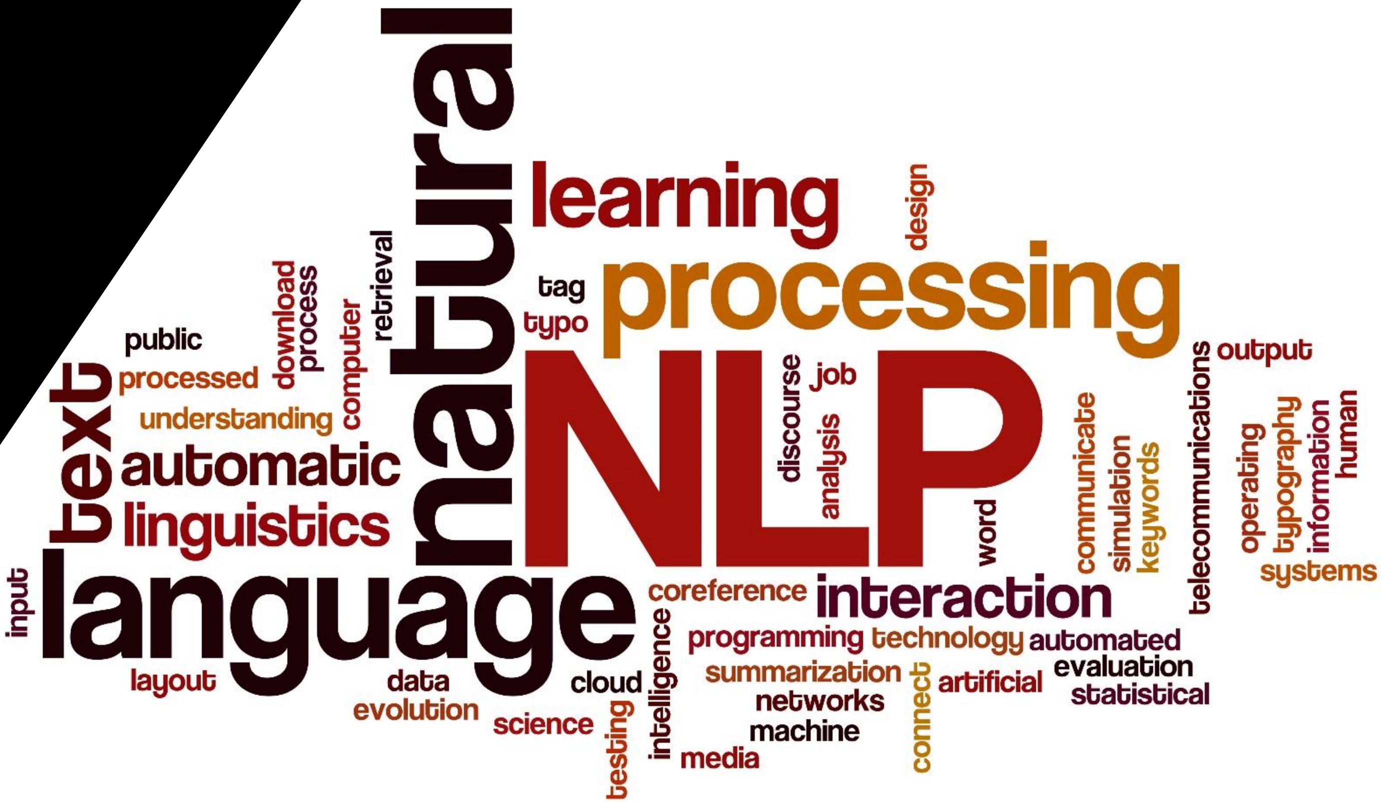
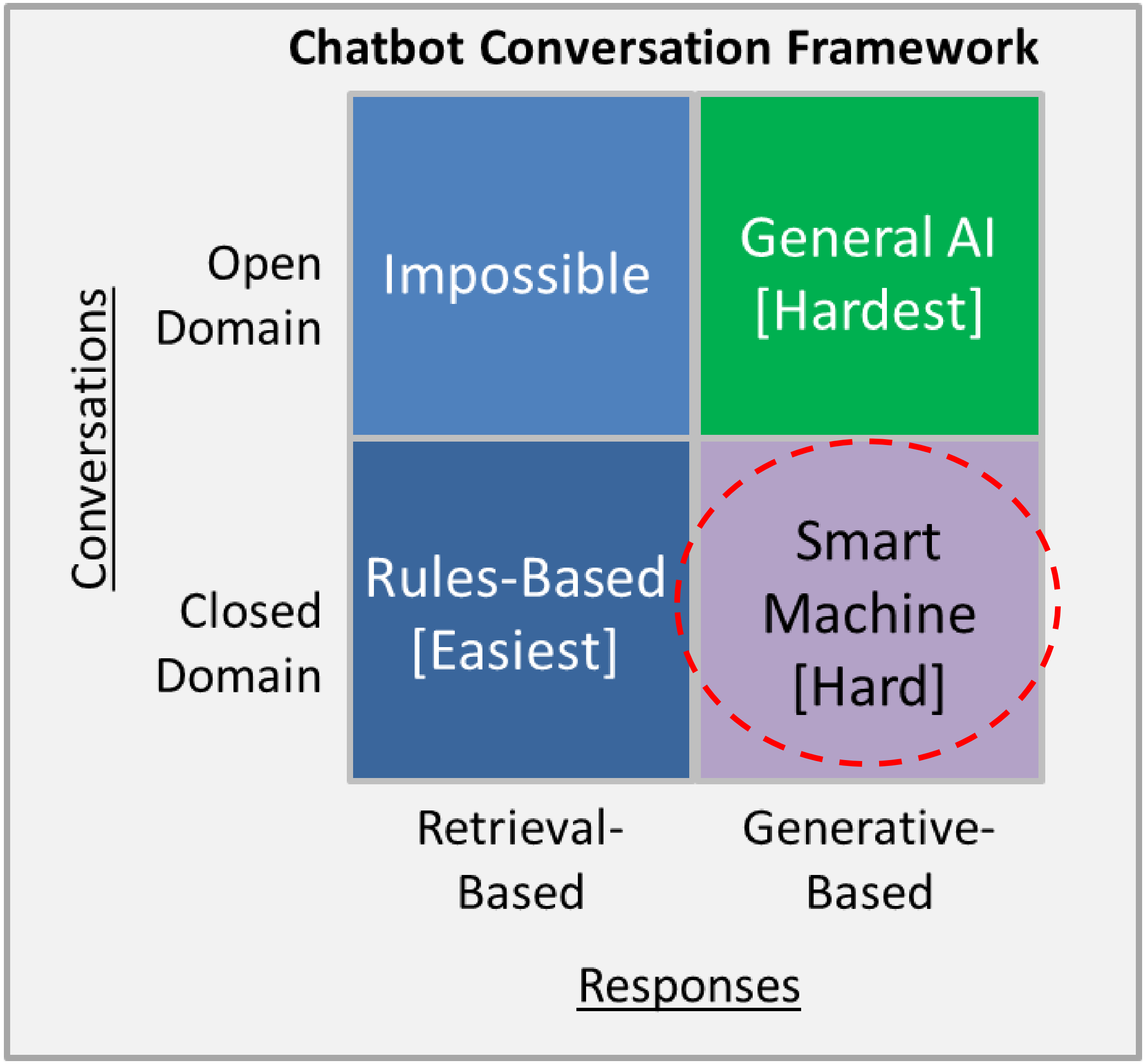
RAONBIT
DATA&SOLUTION

# 딥러닝 자연어처리 기반의 Chatbot 시스템 구현

*AI Summit 2019 Seoul Workshop*



# Chatbot

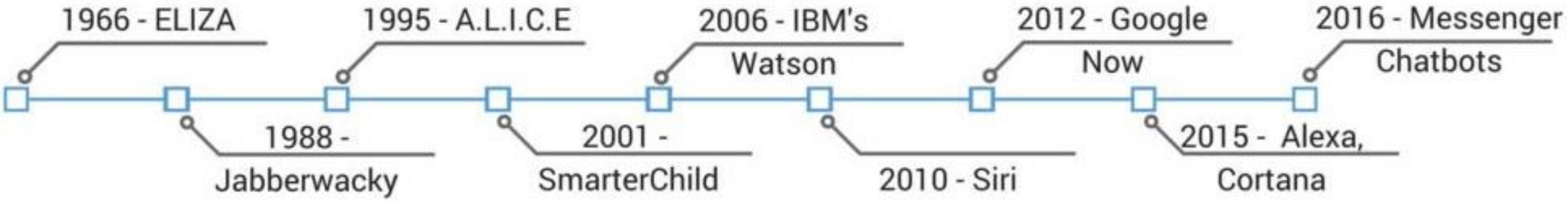


Chatbot Conversation Framework

| Conversations | Retrieval-Based | Generative-Based |
|---|---|---|
| Open Domain | Impossible | General AI [Hardest] |
| Closed Domain | Rules-Based [Easiest] | Smart Machine [Hard] |

Responses

# History



1966 - ELIZA

1988 - Jabberwacky

1995 - A.L.I.C.E

2001 - SmarterChild

2006 - IBM's Watson

2010 - Siri

2012 - Google Now

2015 - Alexa, Cortana

2016 - Messenger Chatbots

- ELIZA : Keyword, Pattern matching, Turing test
- ALICE : NLP, Heuristic pattern matching
- SmarterChild : Intelligentt bot, Siri, S-Voice
- Watson : NLP, Machine learning
- Siri : Personal assistant
- Google Now : Recommendation, Web service
- Alexa : Amazon echo device
- Cortana : MS personal assistant
- Messenger Chatbot : SNS Messenger platform

*https://www.altexsoft.com/blog/business/a-comprehensive-guide-to-chatbots-best-practices-for-building-conversational-interfaces/*

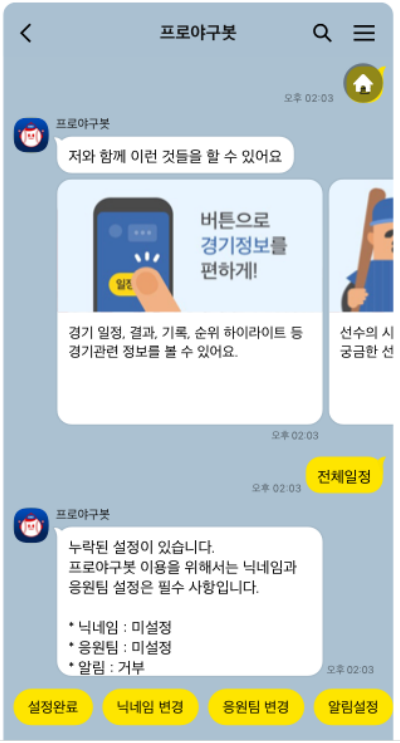
# NLP(Natural Language Processing) Business

- Spell Check & Correction
- Search Autocomplete & Autocorrection
- Smart Search
- Messenger Bots : 주문, 호출 등
- Virtual Assistant : 특정 분야 QnA, A/S 접수
- Knowledge Base Support : 지식 검색
- AI Speaker
- Survey Analytics
- Social Media Monitoring

# NLP Technologies

- 데이터 전처리
  - ✓ Tokenization
  - ✓ Cleaning, Stop word
  - ✓ 형태소 분석
- 문서(문장) 유사도
  - ✓ Cosine similarity
  - ✓ Levenshtein distance
- Topic Modeling
  - ✓ LSA
  - ✓ LDA
- Word Representation
  - ✓ One-hot encoding
  - ✓ Count-Based
    - Bag of Words
    - Document-Term Matrix(DTM)
    - Term-Document Matrix(TDM)
    - TF-IDF
  - ✓ Word Embedding
    - Word2Vec
    - GloVe
    - FastText
    - ELMO
    - BERT

# Chatbot 분류


프로야구봇

오후 02:03

프로야구봇

저와 함께 이런 것들을 할 수 있어요

버튼으로
경기정보를
편하게!

경기 일정, 결과, 기록, 순위 하이라이트 등
경기관련 정보를 볼 수 있어요.

선수의 시
궁금한 선

오후 02:03

전체일정

오후 02:03

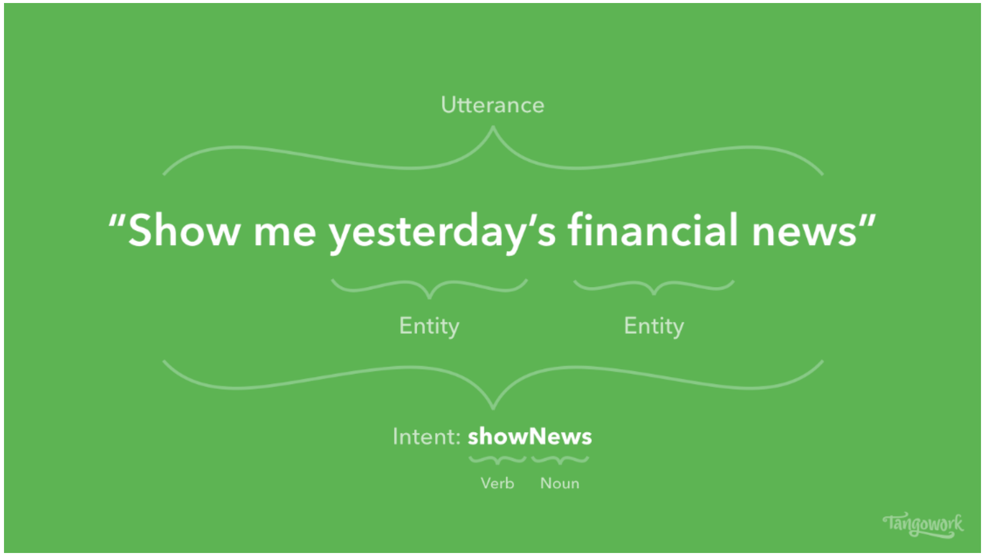
프로야구봇

누락된 설정이 있습니다.
프로야구봇 이용을 위해서는 닉네임과
응원팀 설정은 필수 사항입니다.

* 닉네임 : 미설정
* 응원팀 : 미설정
* 알림 : 거부

오후 02:03

설정완료 | 닉네임 변경 | 응원팀 변경 | 알림설정



Azure
Bot
Service

# Chatbot 고려 사항

### 학습데이터구성

- 데이터 수집/정제
- STT, TTS

### 데이터 전처리

- 단어 구분
- 형태소 분석
- 도메인 사전
- 유의어 사전

### 학습 모델

- Word embedding
- Network 설계
- Error Analysis

### 서비스 시스템

- User Interface
- Serving Architecture
- Multi User
- Response Latency

### 사용자 오류

- 맞춤법 / 띄어쓰기
- 문법 오류

### 데이터 분포

- 인사말/비속어/냉소
- 비학습 데이터
- 구어체 / 문어체
- 신조어 / 약어

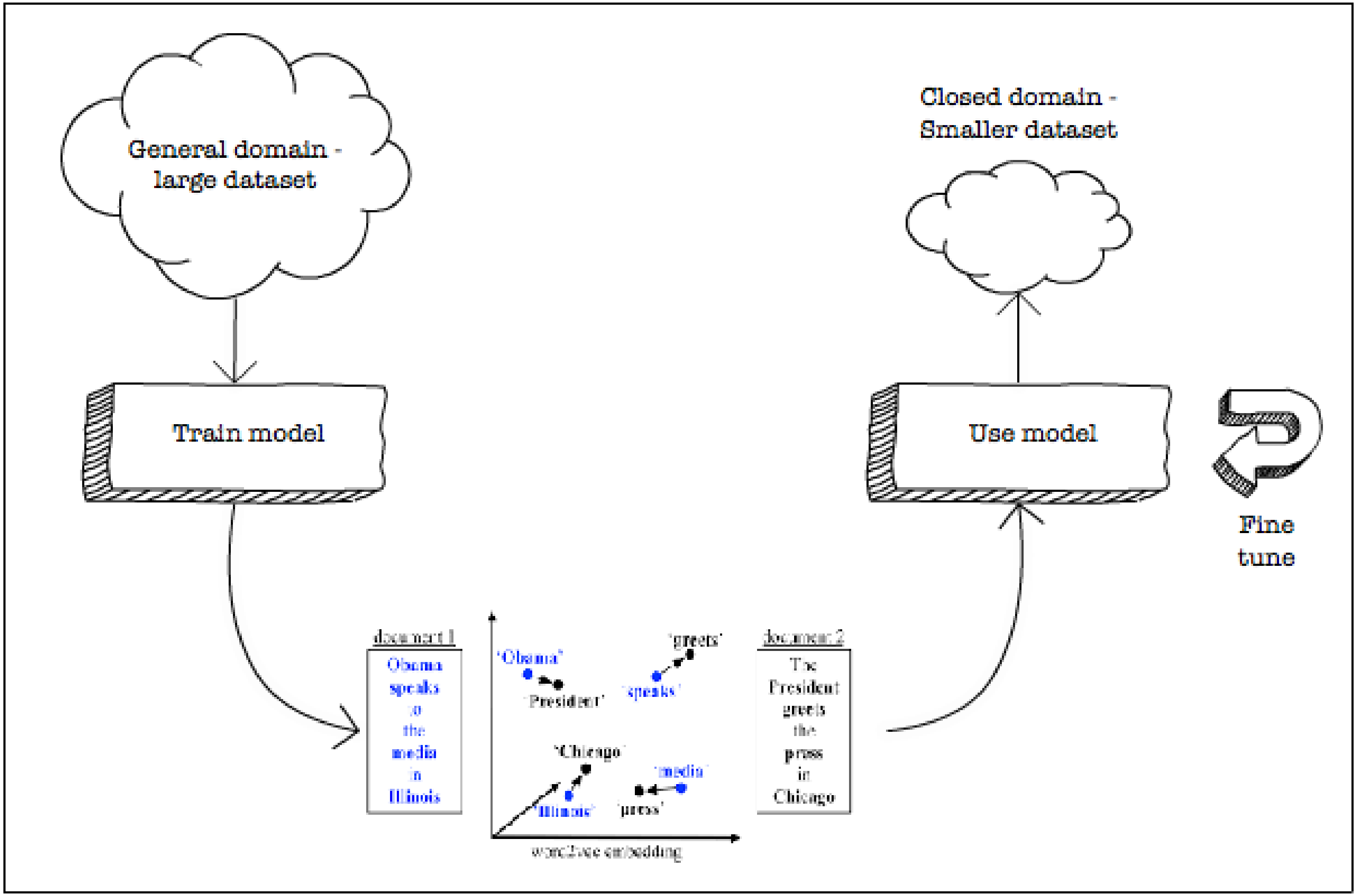
### 이상 답변 처리

- 법률적/의학적 문제
- 비속어 답변
- 문법 오류

### 사용자 FeedBack

- 오류 답변 구분
- 재학습 데이터 구성

# 최신 동향 – Transfer Learning



General domain - large dataset

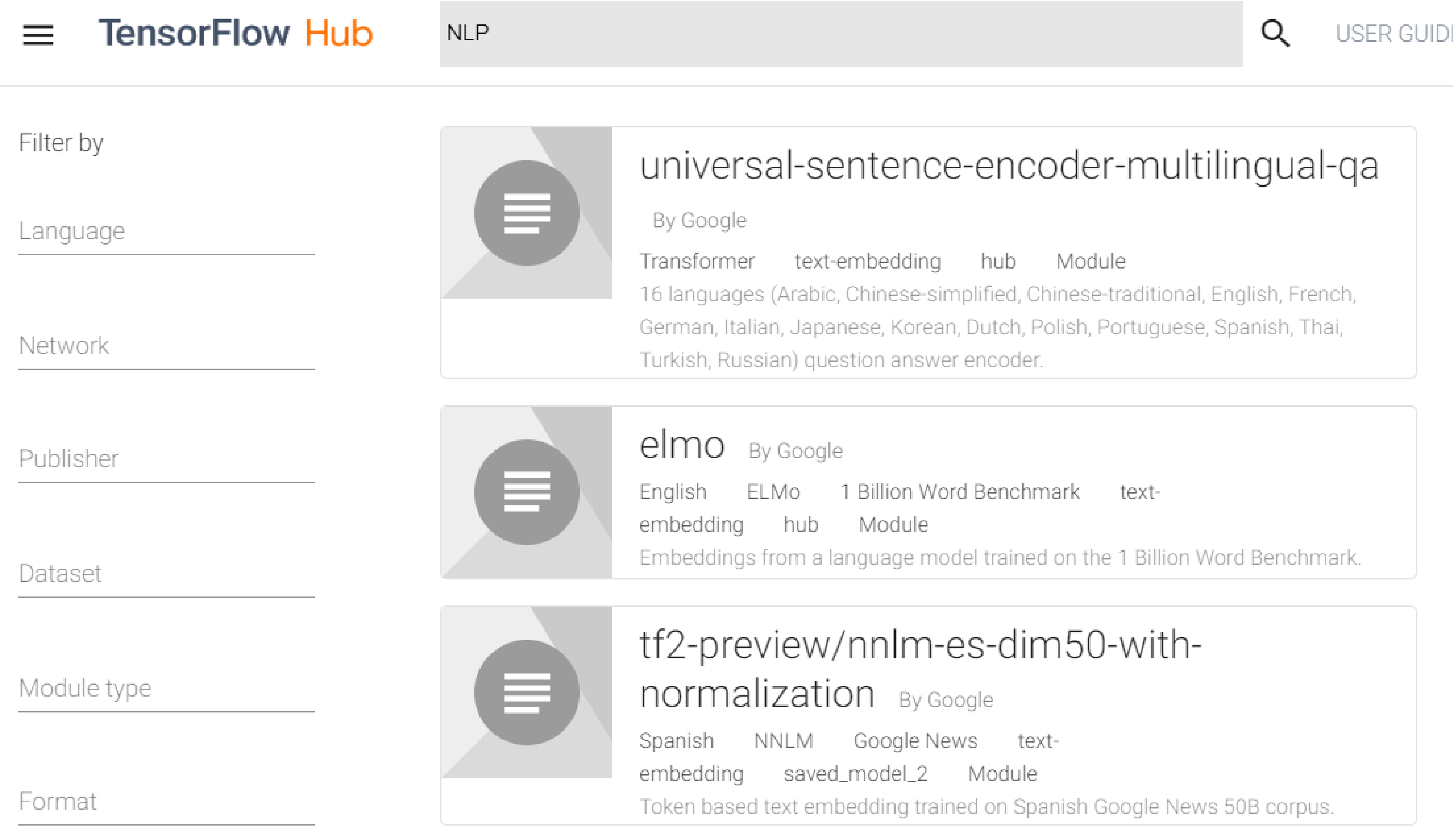
Train model

Closed domain - Smaller dataset

Use model

Fine tune

# 최신 동향 – Pre-trained model



Filter by

Language

Network

Publisher

Dataset

Module type

Format

universal-sentence-encoder-multilingual-qa

By Google

Transformer text-embedding hub Module

16 languages (Arabic, Chinese-simplified, Chinese-traditional, English, French, German, Italian, Japanese, Korean, Dutch, Polish, Portuguese, Spanish, Thai, Turkish, Russian) question answer encoder.

elmo By Google

English ELMo 1 Billion Word Benchmark text-embedding hub Module

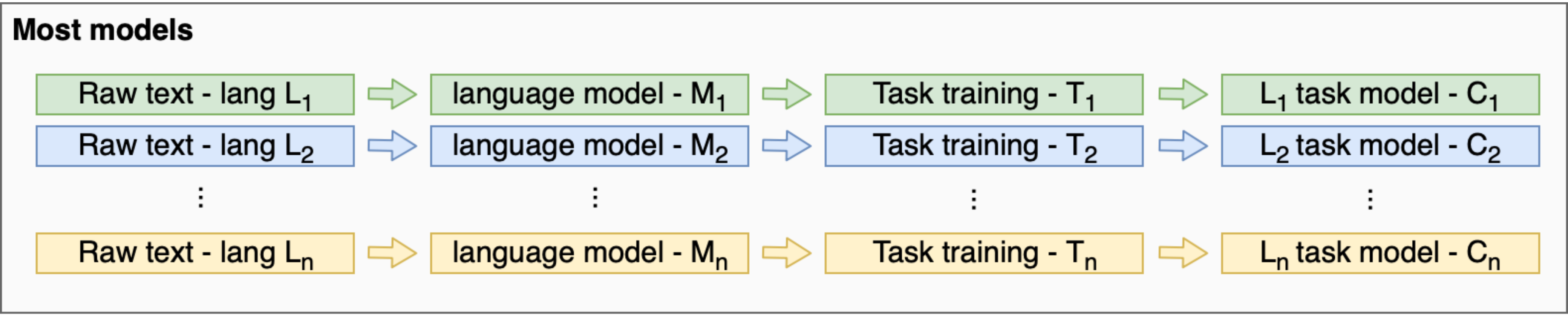
Embeddings from a language model trained on the 1 Billion Word Benchmark.

tf2-preview/nnlm-es-dim50-with-normalization By Google

Spanish NNLM Google News text-embedding saved_model_2 Module

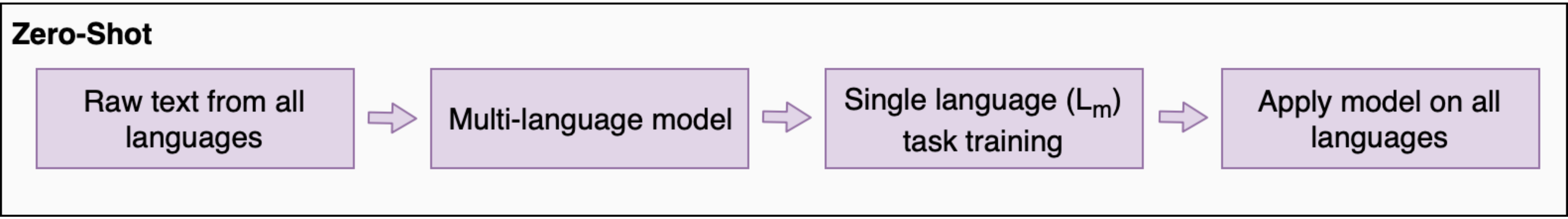
Token based text embedding trained on Spanish Google News 50B corpus.

# 최신 동향 – Zero shot learning



**Most models**

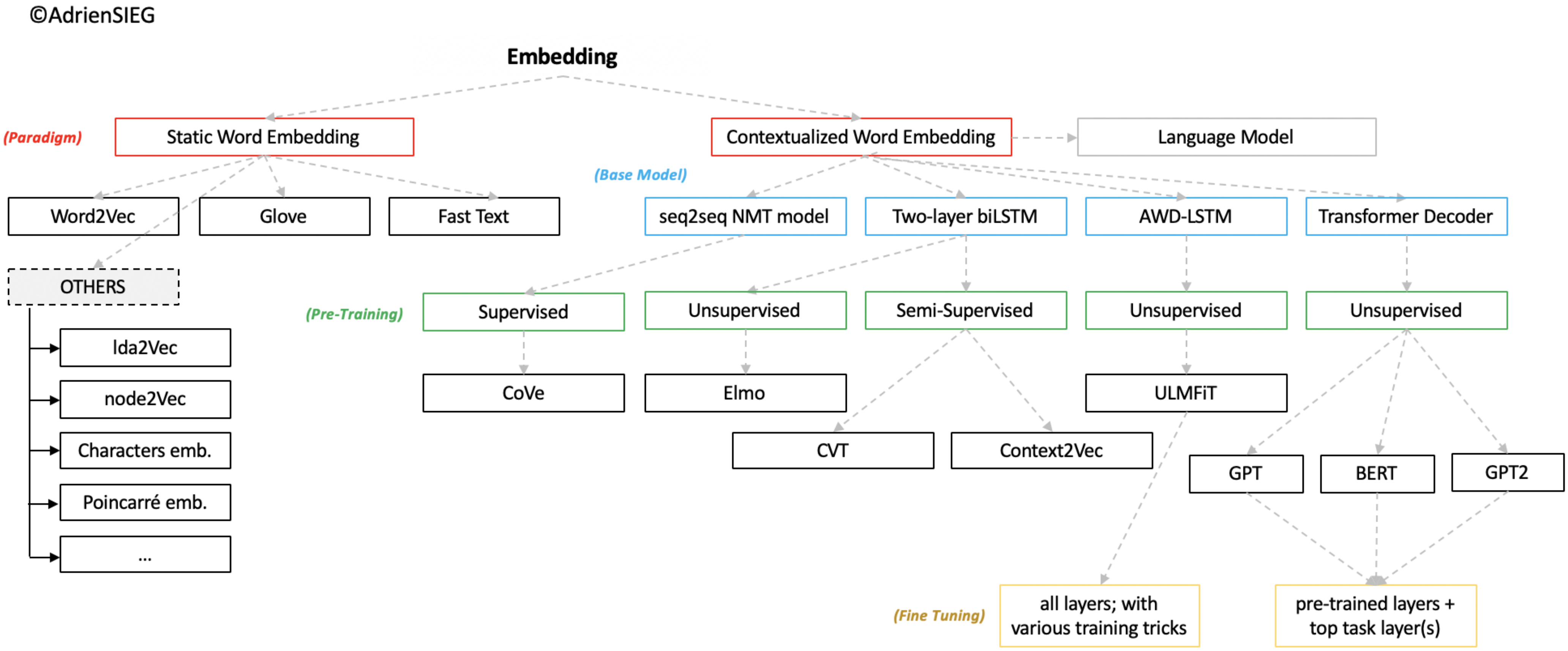
| | | | |
|---|---|---|---|
| Raw text - lang $L_1$ | language model - $M_1$ | Task training - $T_1$ | $L_1$ task model - $C_1$ |
| Raw text - lang $L_2$ | language model - $M_2$ | Task training - $T_2$ | $L_2$ task model - $C_2$ |
| ⋮ | ⋮ | ⋮ | ⋮ |
| Raw text - lang $L_n$ | language model - $M_n$ | Task training - $T_n$ | $L_n$ task model - $C_n$ |



**Zero-Shot**

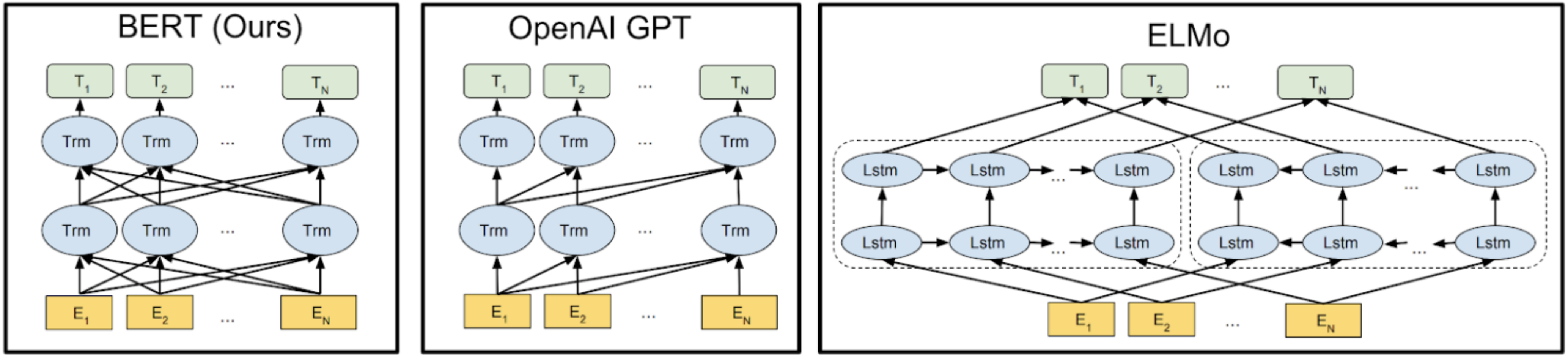
Raw text from all languages → Multi-language model → Single language ($L_m$) task training → Apply model on all languages

# Word Embedding



https://miro.medium.com/max/2438/1*ff_bprXLuTueAx7-5-MHew.png

# Word Embedding



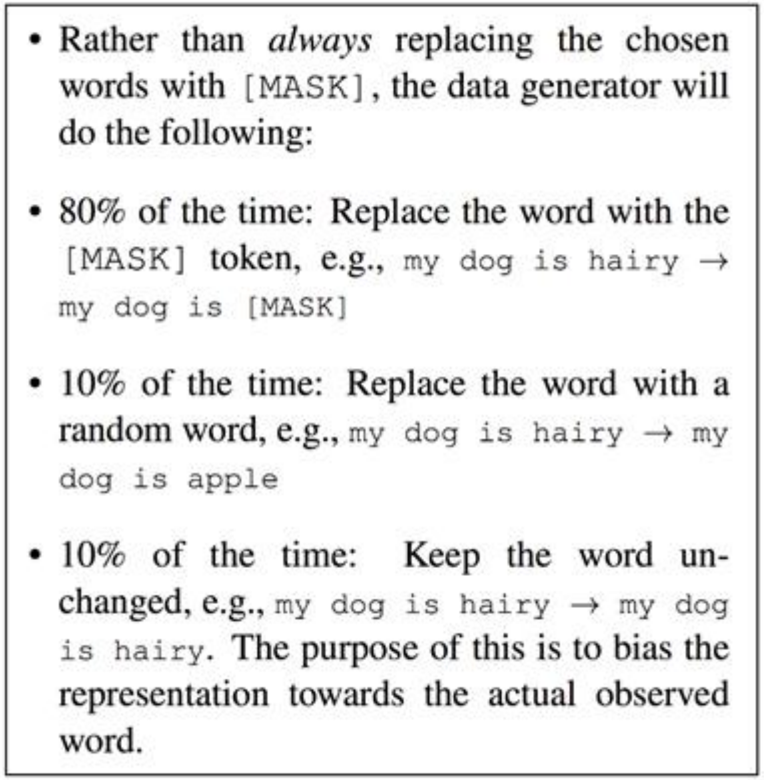
- ELMo : 순방향 Encoder(LSTM)과 역방향 Decoder(LSTM)을 사용하여 학습

- GPT : 순방향 Transformer를 사용하여 학습

- BERT : 양방향 Transformer를 사용하여 학습

*https://arxiv.org/abs/1810.04805*

# Word Embedding

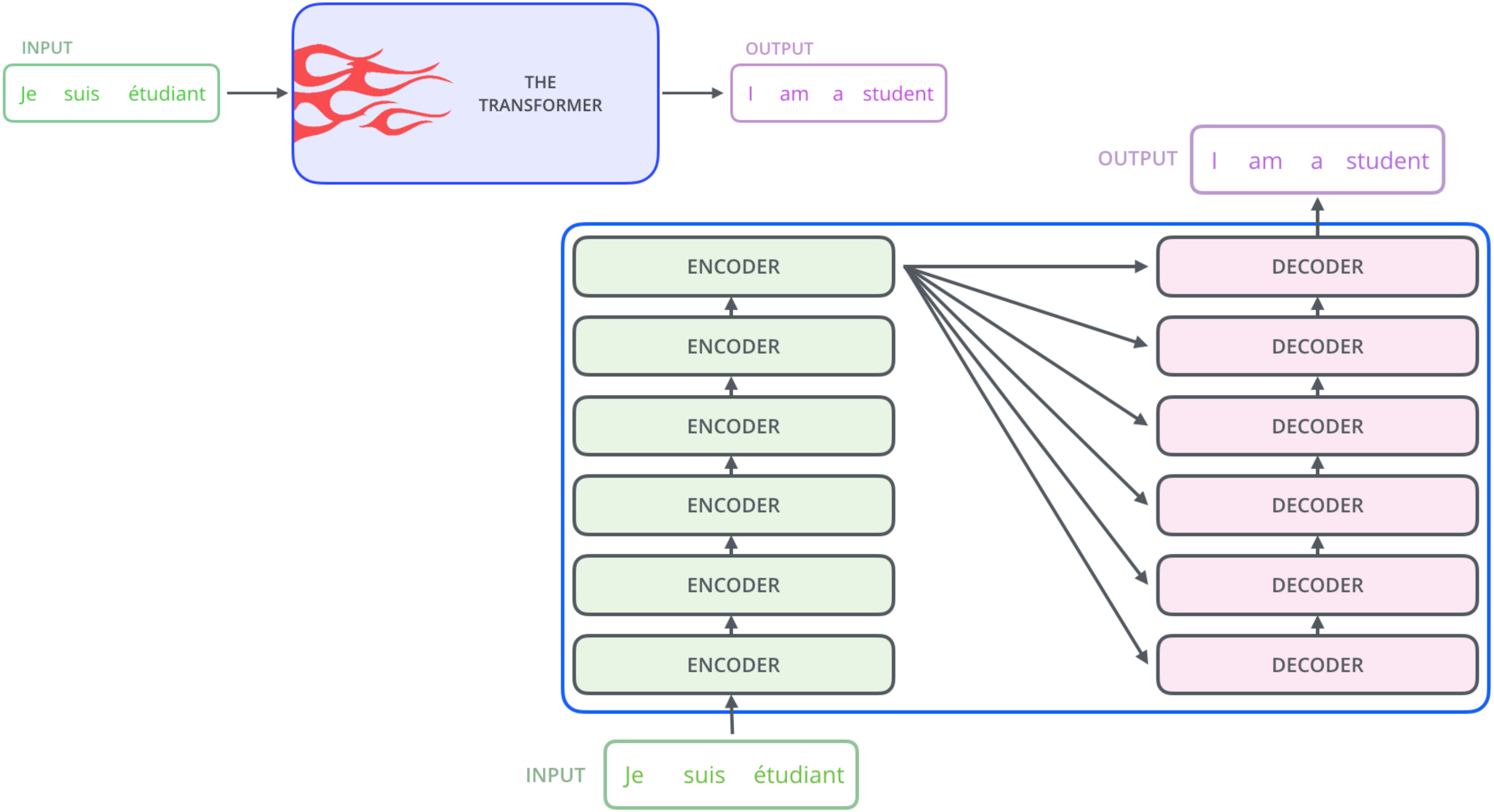


- Rather than *always* replacing the chosen words with [MASK], the data generator will do the following:
- 80% of the time: Replace the word with the [MASK] token, e.g., my dog is hairy → my dog is [MASK]
- 10% of the time: Replace the word with a random word, e.g., my dog is hairy → my dog is apple
- 10% of the time: Keep the word unchanged, e.g., my dog is hairy → my dog is hairy. The purpose of this is to bias the representation towards the actual observed word.

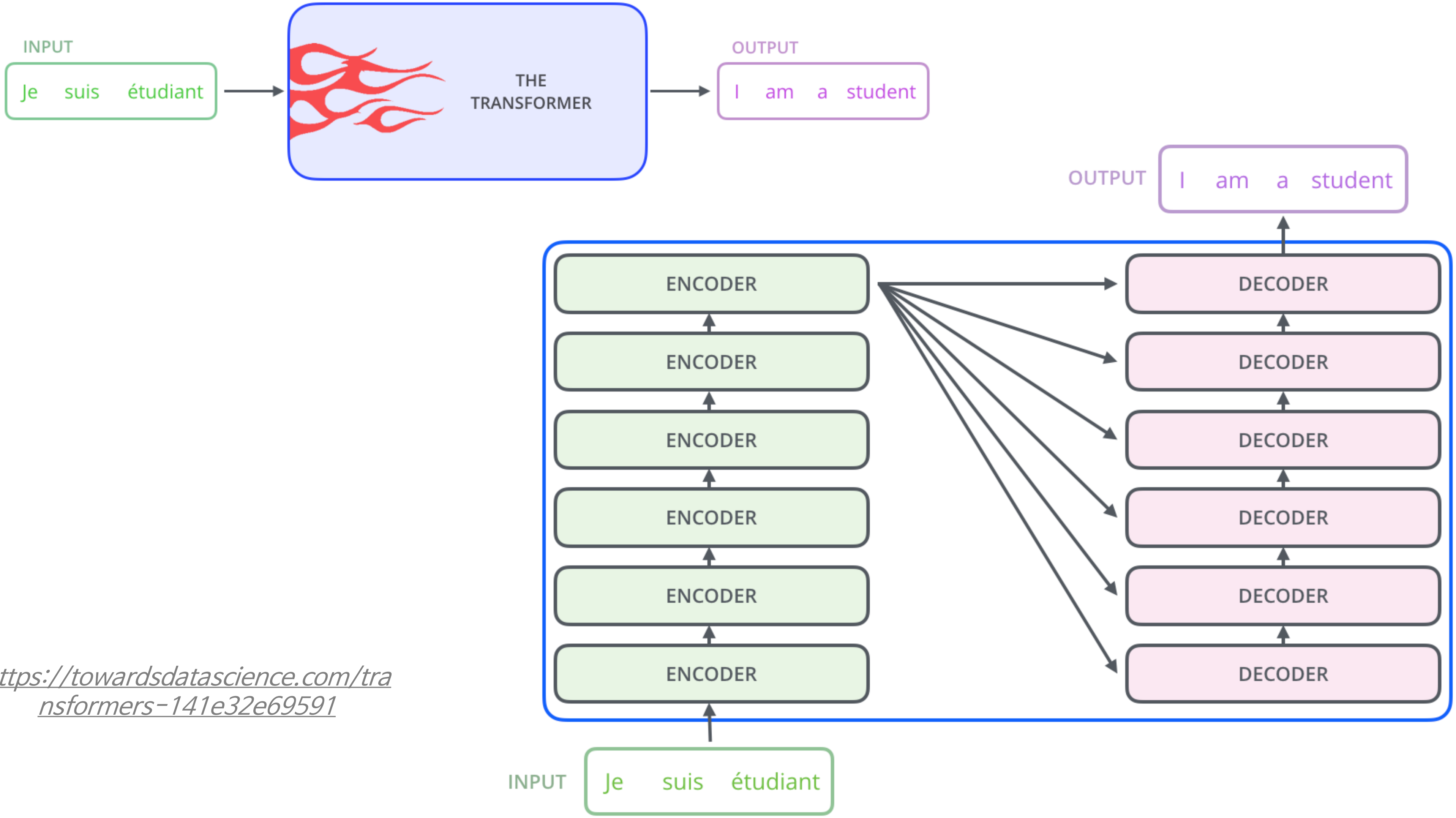
- Mark Language Model : 입력 값 왜곡

입력 단어 배열에서 무작위적으로 단어를 선택해서 80%의 확률로 [MASK] 로 치환하거나 10%의 확률로 임의의 단어로 치환하거나 나머지 10%의 확률로 바꾸지 않고 그대로 사용

# Transformer

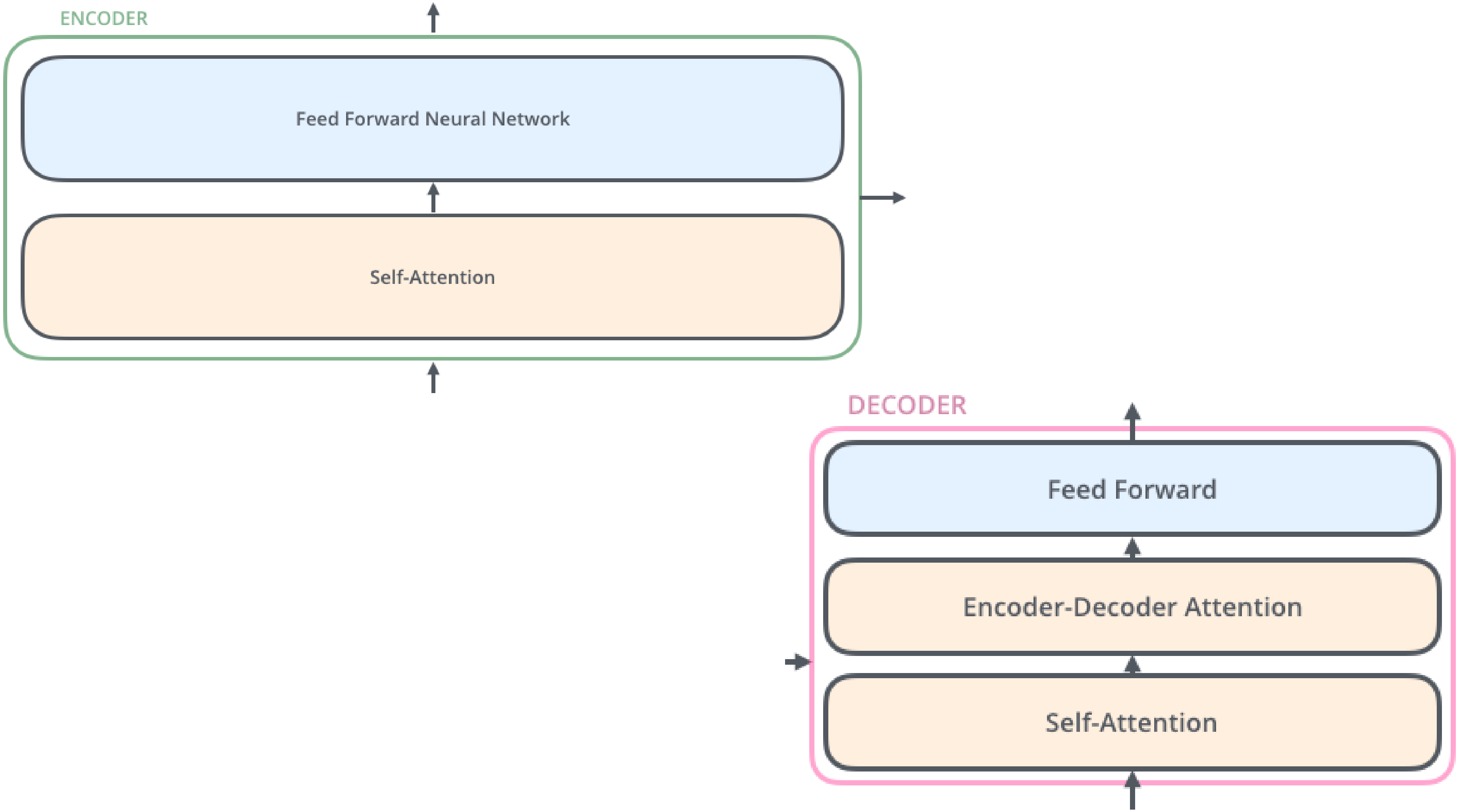


# Transformer

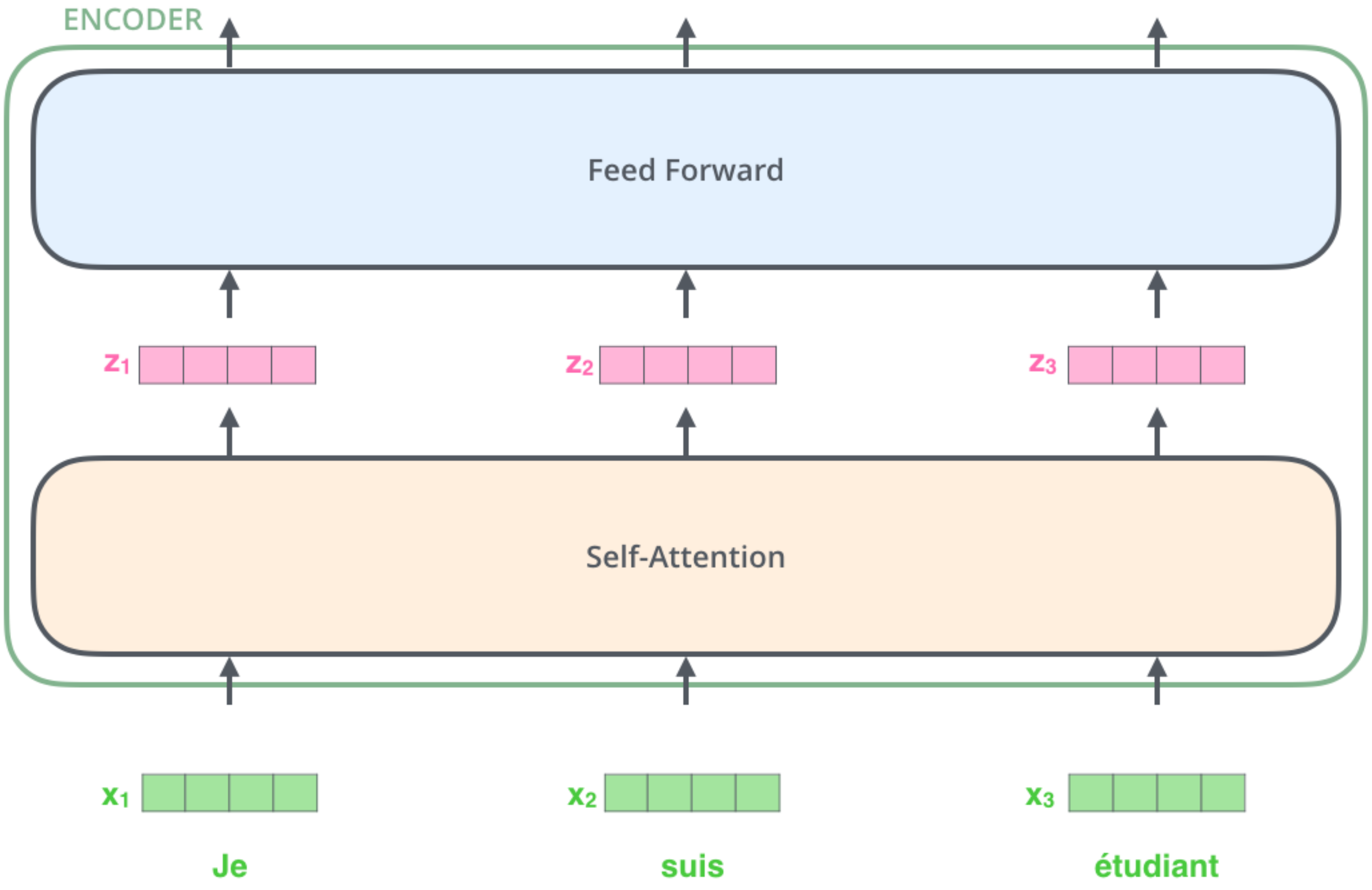


https://towardsdatascience.com/transformers-141e32e69591

# Transformer

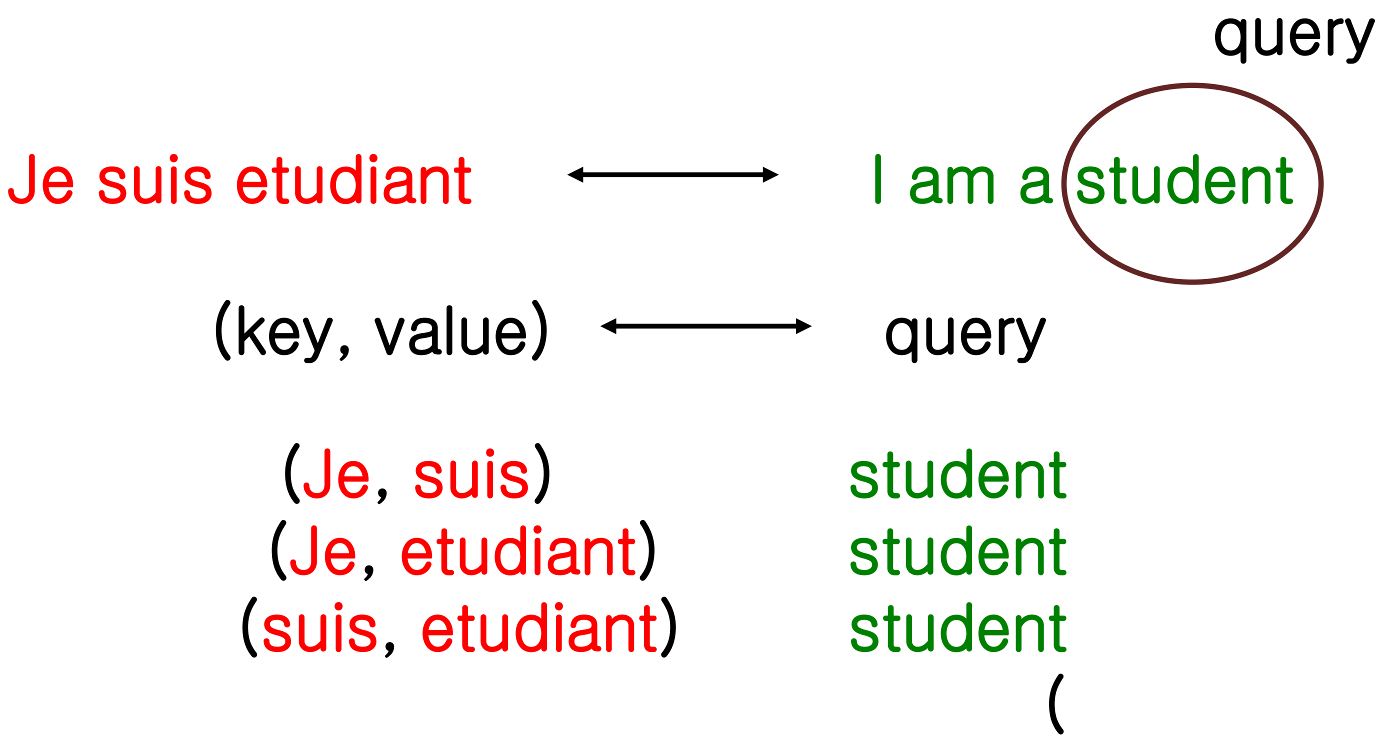


# Transformer



Self-Attention

# Transformer



query

Je suis etudiant ⟷ I am a student

(key, value) ⟷ query

(Je, suis) student

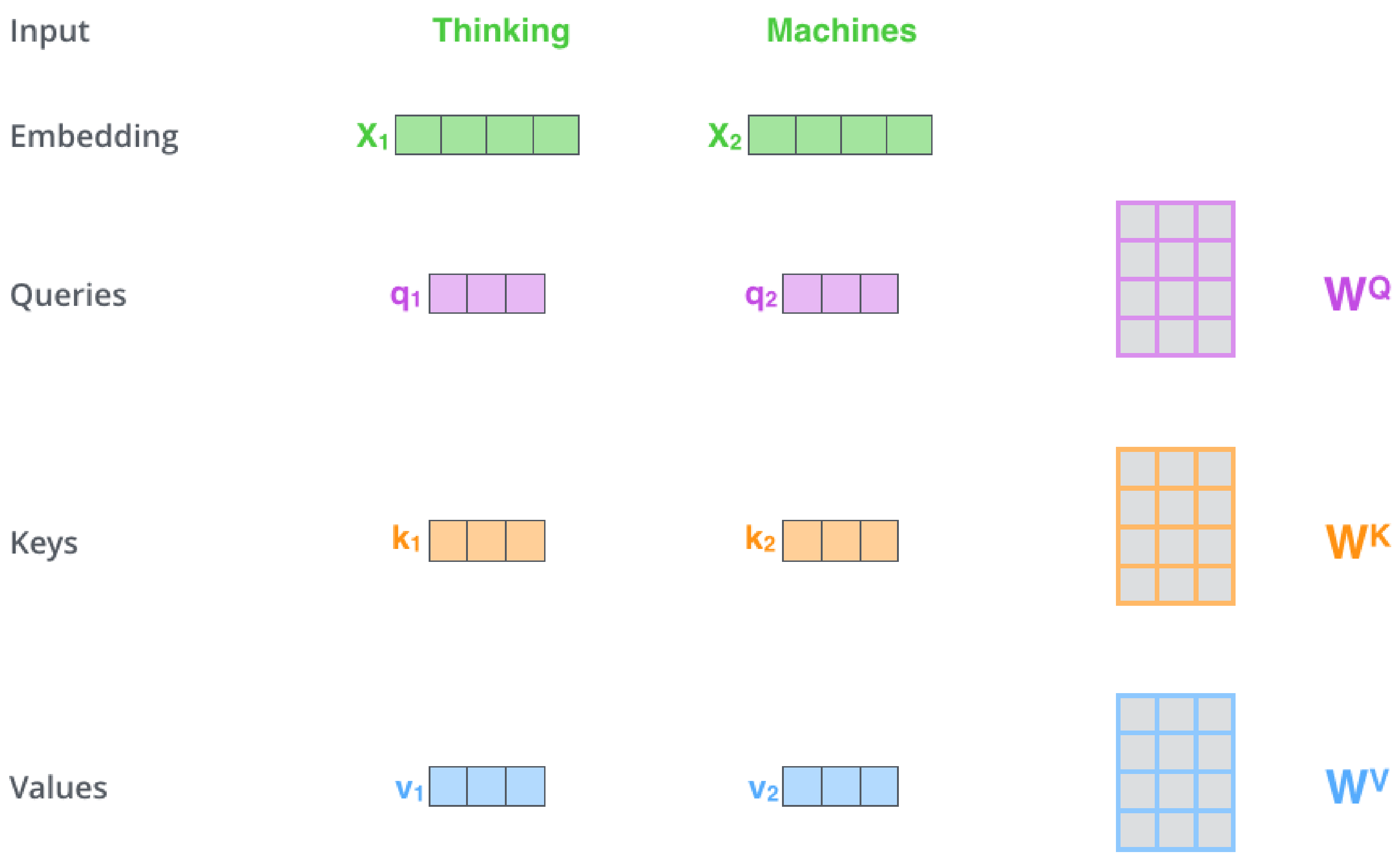
(Je, etudiant) student

(suis, etudiant) student

(

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

# Transformer



Input Thinking Machines

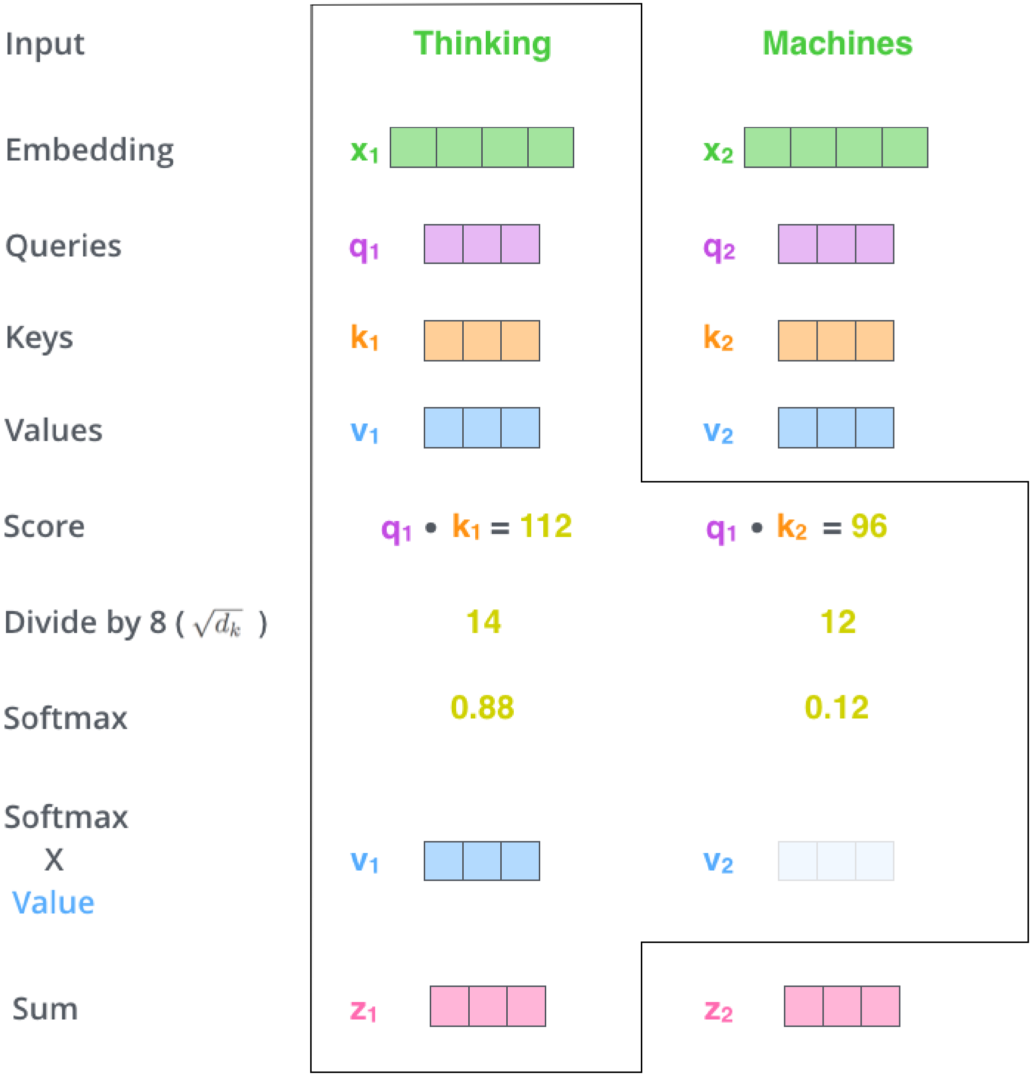
Embedding $X_1$ $X_2$

Queries $q_1$ $q_2$ $W^Q$

Keys $k_1$ $k_2$ $W^K$

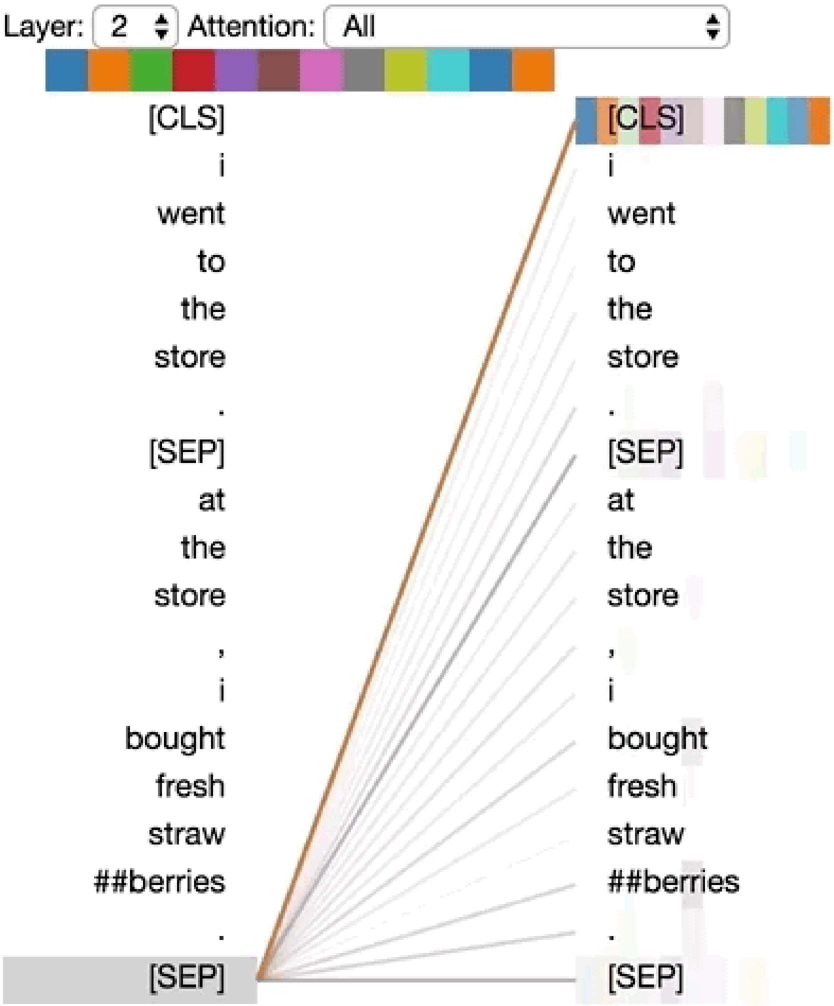
Values $v_1$ $v_2$ $W^V$

# Transformer



| Input | Thinking | Machines |
|---|---|---|
| Embedding | $x_1$ | $x_2$ |
| Queries | $q_1$ | $q_2$ |
| Keys | $k_1$ | $k_2$ |
| Values | $v_1$ | $v_2$ |
| Score | $q_1 \bullet k_1 = 112$ | $q_1 \bullet k_2 = 96$ |
| Divide by 8 ( $\sqrt{d_k}$ ) | 14 | 12 |
| Softmax | 0.88 | 0.12 |
| Softmax<br>X<br>Value | $v_1$ | $v_2$ |
| Sum | $z_1$ | $z_2$ |

# Transformer



Layer: 2 Attention: All

# Transformer

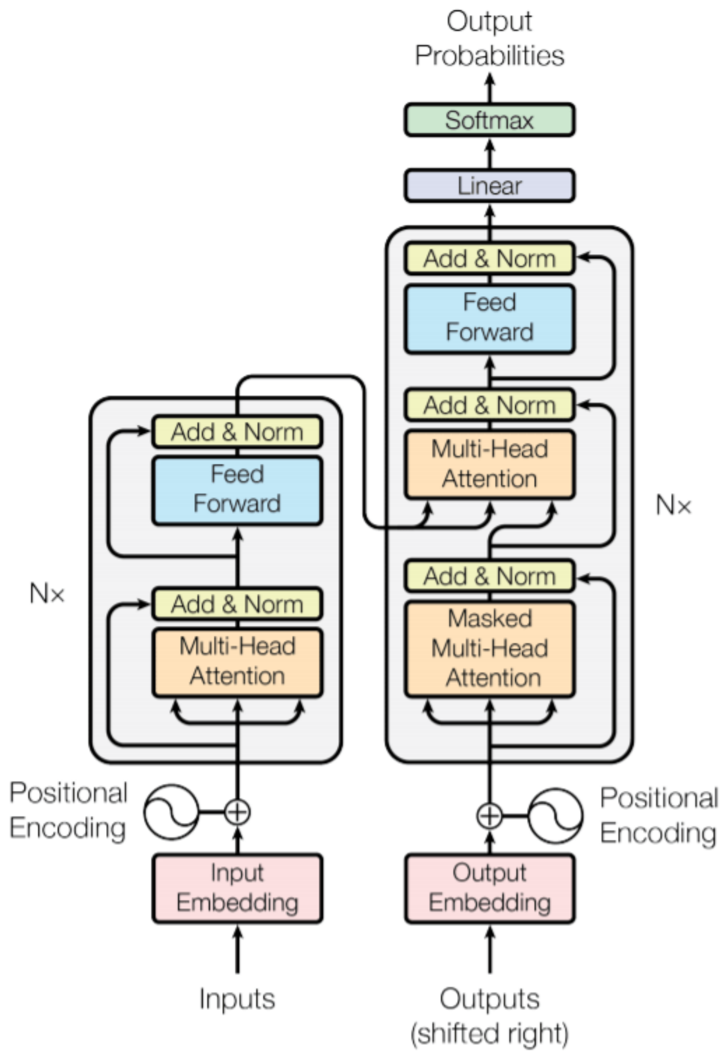
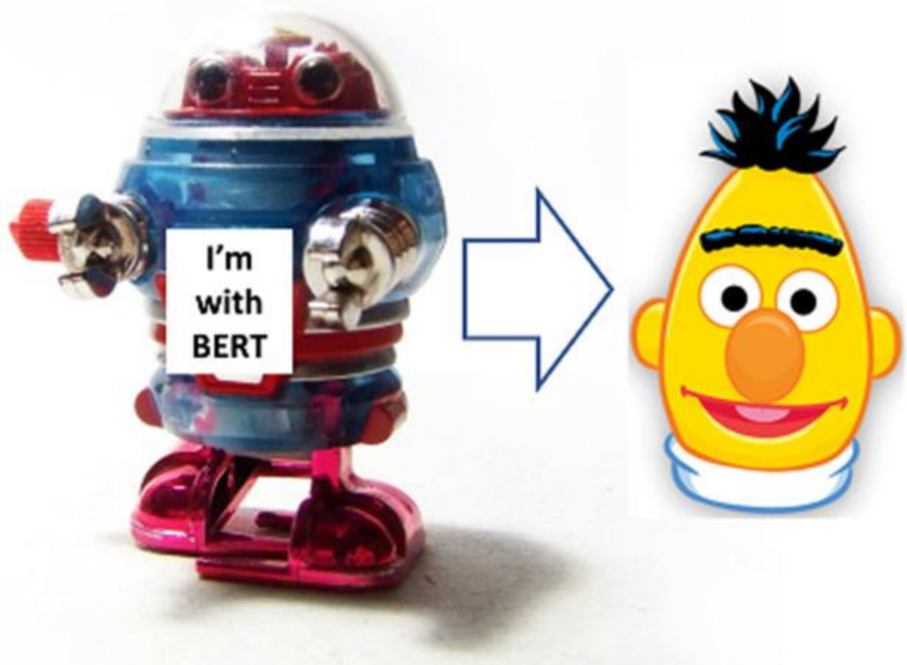


Figure 1: The Transformer - model architecture.

# BERT(Bidirectional Encoder Representations form Transformer)



- Transformer를 활용한 Language Representation Model
- 기본적으로, wiki와 같은 대용량 unlabeled data로 모델을 미리 학습 시킨 후, 특정 *task*를 가지고 있는 *labeled data*로 transfer learning을 하는 모델
- ELMo, GPT의 단점을 개선
- BERT 모델 자체의 fine-tuning을 통해 활용 가능

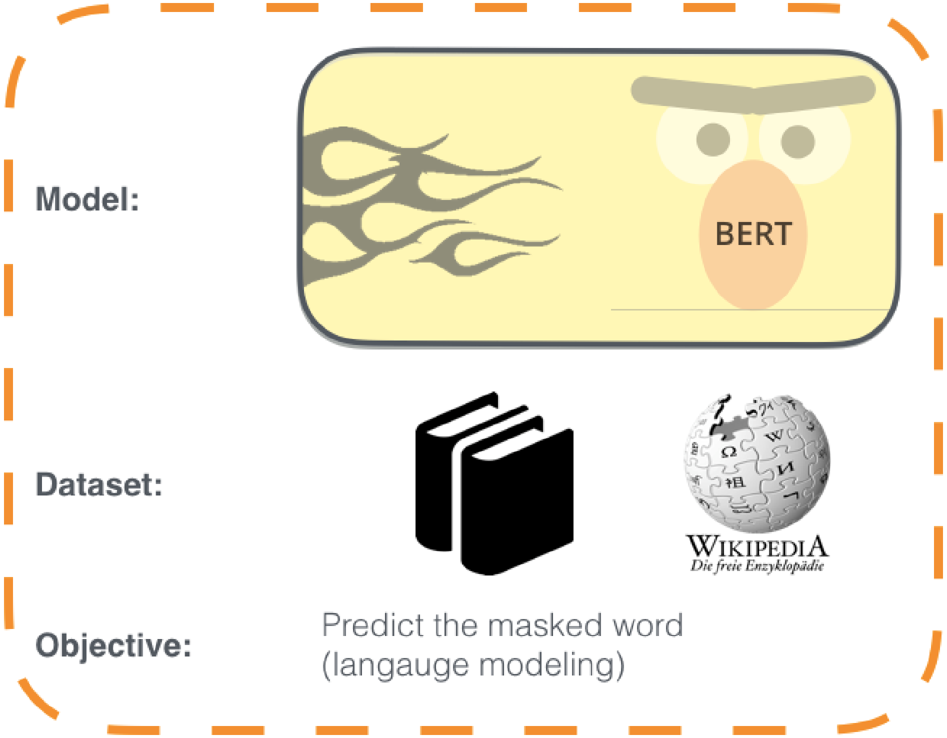
- feature-based approach : 특정 task를 수행하는 network에 pre-trained language representation을 추가적인 feature로 제공. 즉, 두 개의 network를 붙여서 사용(ELMo)
- fine-tuning approach : pre-trained된 parameter들을 일부만 변경하여 사용하는 방식 (OpenAI GPT, BERT)

# BERT(Bidirectional Encoder Representations form Transformer)

## 1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

### Semi-supervised Learning Step



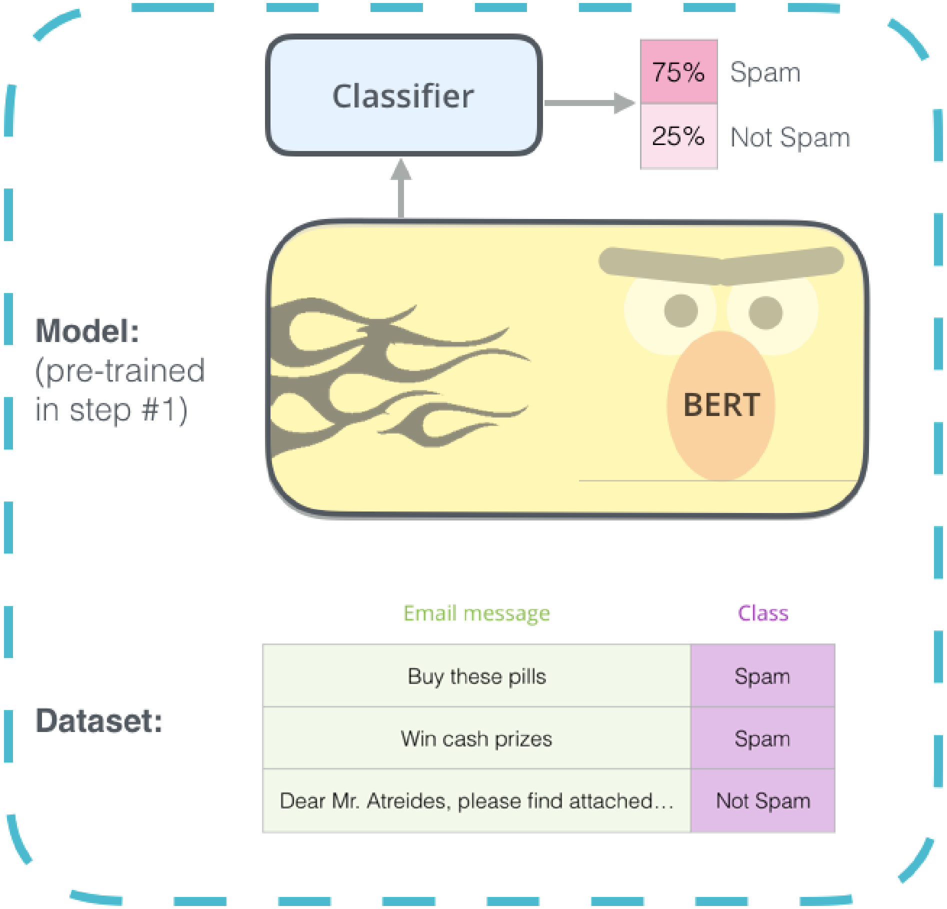
**Model:**

**Dataset:**

**Objective:** Predict the masked word (langauge modeling)

## 2 - Supervised training on a specific task with a labeled dataset.

### Supervised Learning Step



**Model:** (pre-trained in step #1)

**Dataset:**

| Email message | Class |
|---|---|
| Buy these pills | Spam |
| Win cash prizes | Spam |
| Dear Mr. Atreides, please find attached… | Not Spam |

# BERT(Bidirectional Encoder Representations form Transformer)

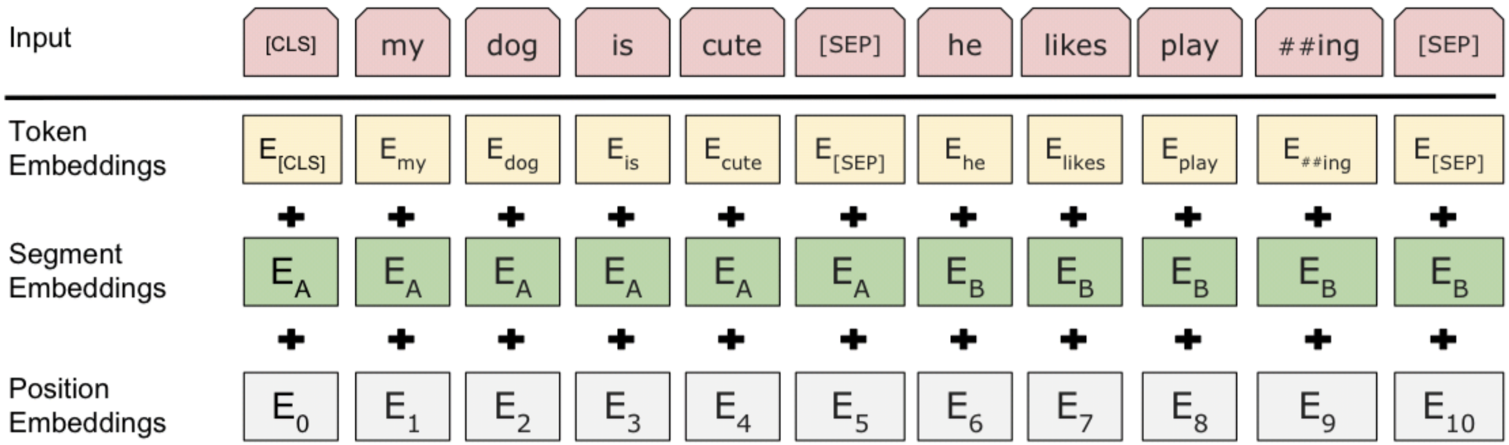
- Model Architecture
  - ✓ *transformer* 의 encoder 부분만 사용
  - ✓ *base* 모델과 *large* 모델을 제공
    - ✓ BERT_base : L=12, H=768, A=12, Total Parameters = 110M
    - ✓ BERT_large : L=24, H=1024, A=16, Total Parameters = 340M

    ( L : transformer block의 layer 수, H : hidden size, A : self-attention heads 수)
  - ✓ OpenAI GPT모델과 *hyper parameter*가 동일
  - ✓ OpenAI GPT모델은 next token 만을 예측하는 기본적인 *language model* 방식을 사용하였고, 그를 위해 transformer decoder를 사용, BERT는 MLM과 NSP를 위해 self-attention을 수행하는 transformer encoder구조를 사용

    (MLM : Masked Language Model, NSP : Next Sentence Prediction)

# BERT(Bidirectional Encoder Representations form Transformer)

- Input representation



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | [CLS] | my | dog | is | cute | [SEP] | he | likes | play | ##ing | [SEP] |
| Token Embeddings | $E_{[CLS]}$ | $E_{my}$ | $E_{dog}$ | $E_{is}$ | $E_{cute}$ | $E_{[SEP]}$ | $E_{he}$ | $E_{likes}$ | $E_{play}$ | $E_{\#\#ing}$ | $E_{[SEP]}$ |
| | + | + | + | + | + | + | + | + | + | + | + |
| Segment Embeddings | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_B$ | $E_B$ | $E_B$ | $E_B$ | $E_B$ |
| | + | + | + | + | + | + | + | + | + | + | + |
| Position Embeddings | $E_0$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_5$ | $E_6$ | $E_7$ | $E_8$ | $E_9$ | $E_{10}$ |

- ✓ [CLS] token : 문장 시작, [SEP] token : 문장 구분
- ✓ Segment Embeddings : 여러 문장의 embedding

# BERT(Bidirectional Encoder Representations form Transformer)

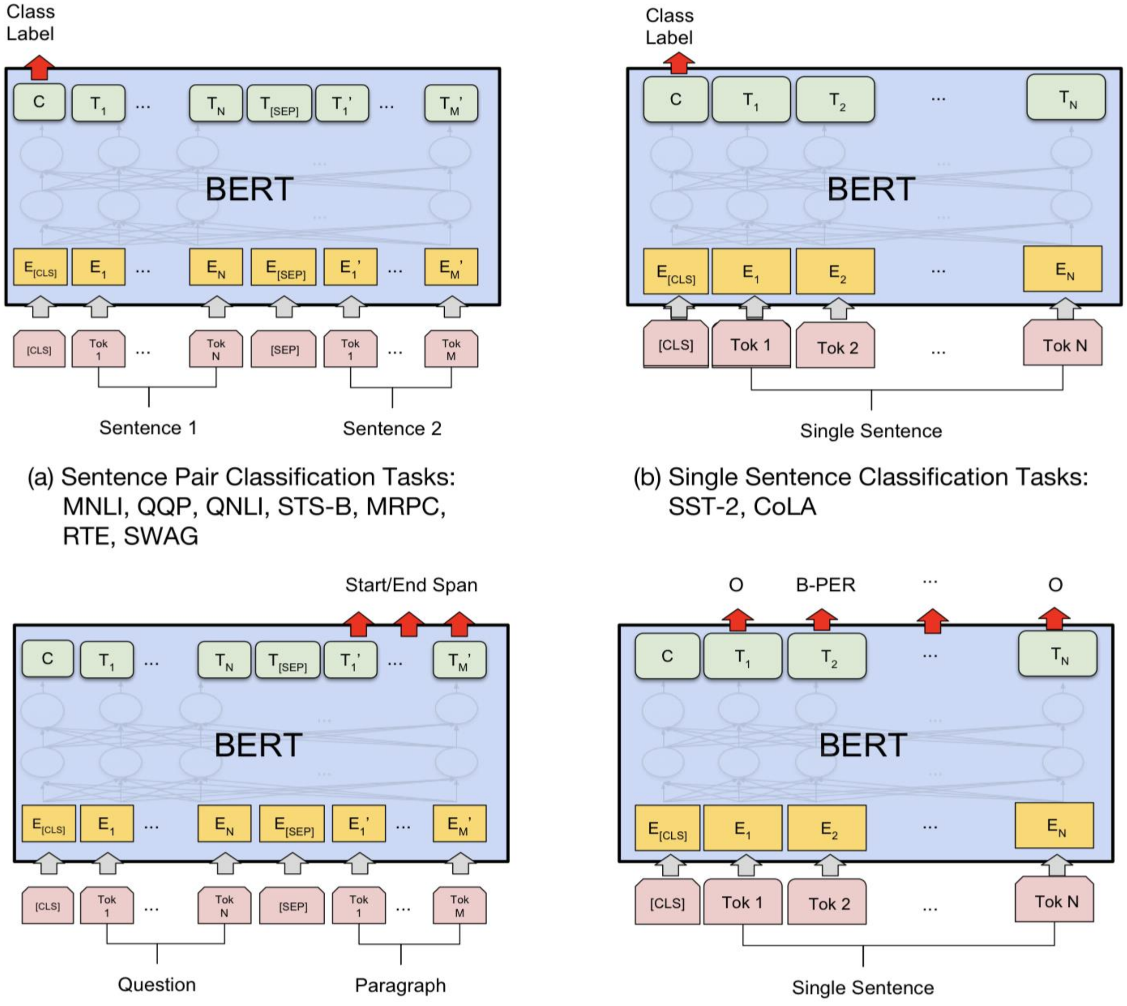
- Training Task – MLM(Masked Language Model)
  - ✓ 단어 중의 일부를 [MASK] token 으로 변경(15%)
    - ➢ 80% : my dog is hairy -> my dog is [MASK]
    - ➢ 10% : my dog is hariy -> my dog is apple
    - ➢ 10% : my dog is hariy -> my dog is hariy
  - ✓ [MASK] token 만을 predict하는 pre-training task를 수행
  - ✓ [MASK] token은 pre-training에만 사용되고, fine-tuning시에는 사용되지 않음. 해당 token을 맞추어 내는 task를 수행하면서, 문맥을 파악하도록 학습
  - ✓ Transformer encoder는 그냥 모든 token에 대해서 distributional contextual representation을 유지하도록 강제
  - ✓ random word로 바꾸는 것은 1.5%(15%의 10%)에 불과하므로 모델의 language understanding 능력에는 지장을 주지 않음

# BERT(Bidirectional Encoder Representations form Transformer)

- Training Task - NSP(Next Sentence Prediction)
  - ✓ QA나 Natural Language Inference(NLI)와 같이 두 문장 사이의 관계를 이해하는 데 활용
  - ✓ corpus에서 두 문장을 이어 붙여 이것이 원래의 corpus에서 바로 이어 붙여져 있던 문장인지를 맞추는 binarized next sentence prediction task를 수행
  - ✓ 50% : sentence A, B가 실제 next sentence
  - ✓ 50% : sentence A, B가 corpus에서 random으로 뽑힌(관계가 없는) 두 문장

# BERT(Bidirectional Encoder Representations form Transformer)

- Fine-tuning



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

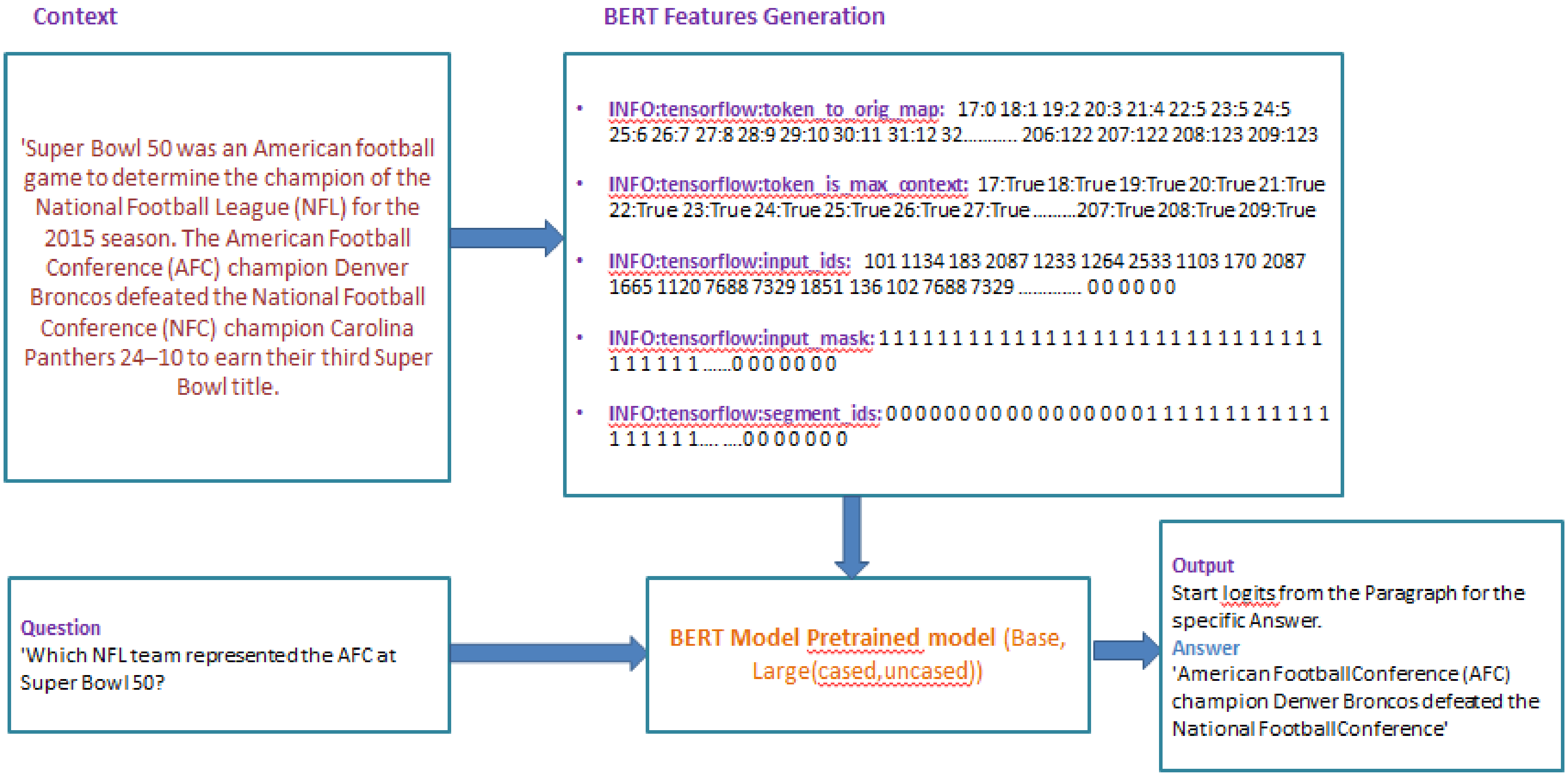
(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

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

# BERT(Bidirectional Encoder Representations form Transformer)

## BERT in Question and Answering



Context

BERT Features Generation

'Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title.

- INFO:tensorflow:token_to_orig_map: 17:0 18:1 19:2 20:3 21:4 22:5 23:5 24:5 25:6 26:7 27:8 28:9 29:10 30:11 31:12 32.......... 206:122 207:122 208:123 209:123
- INFO:tensorflow:token_is_max_context: 17:True 18:True 19:True 20:True 21:True 22:True 23:True 24:True 25:True 26:True 27:True .........207:True 208:True 209:True
- INFO:tensorflow:input_ids: 101 1134 183 2087 1233 1264 2533 1103 170 2087 1665 1120 7688 7329 1851 136 102 7688 7329 ............ 0 0 0 0 0 0
- INFO:tensorflow:input_mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ......0 0 0 0 0 0 0
- INFO:tensorflow:segment_ids: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1.... ....0 0 0 0 0 0 0

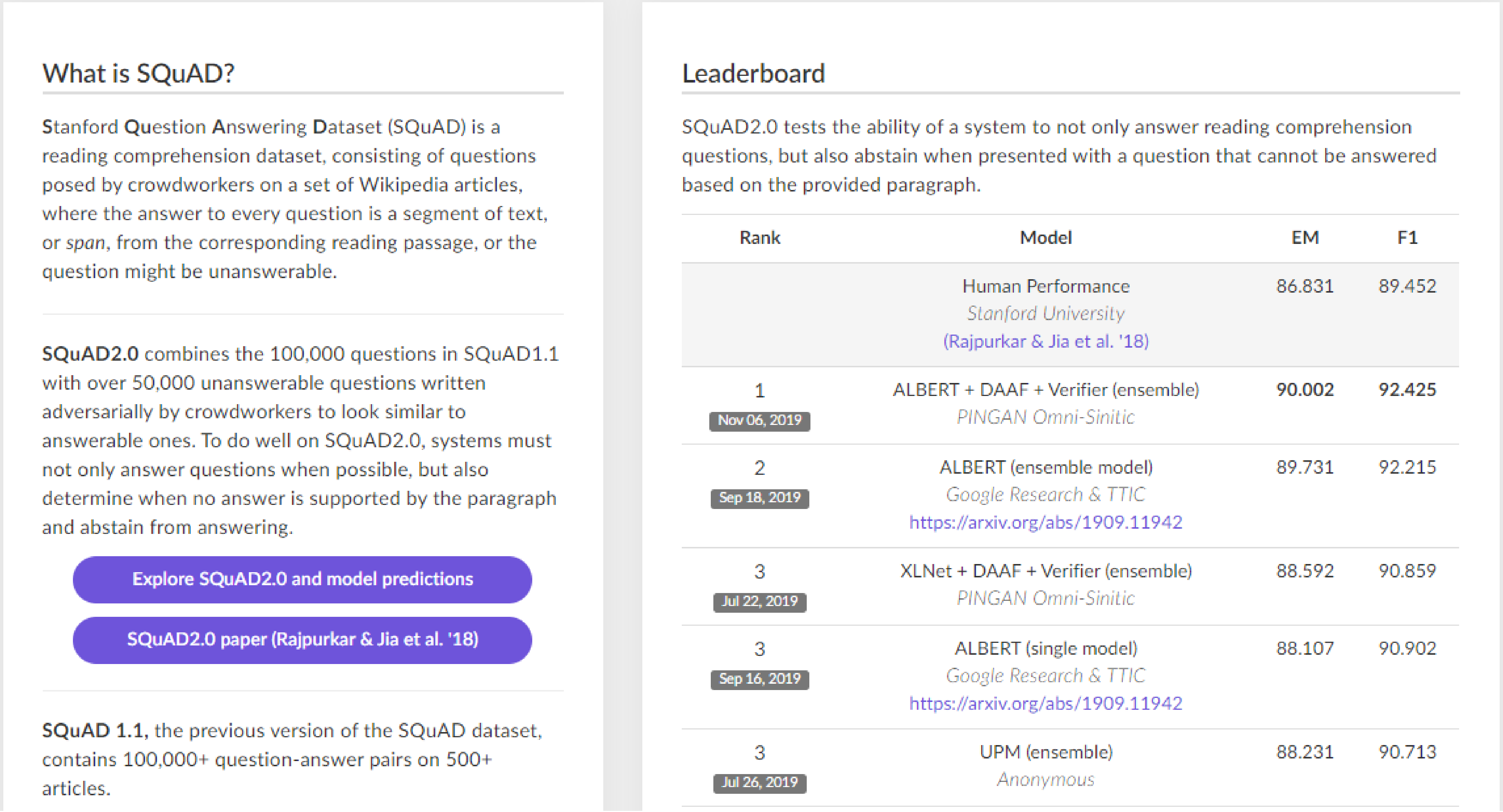
Question
'Which NFL team represented the AFC at Super Bowl 50?

BERT Model Pretrained model (Base, Large(cased,uncased))

Output
Start logits from the Paragraph for the specific Answer.
Answer
'American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference'

https://medium.com/datadriveninvestor/extending-google-bert-as-question-and-answering-model-and-chatbot-e3e7b47b721a

# SQuAD(Stanford Question Answering Dataset)



## What is SQuAD?

**S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

**SQuAD2.0** combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

Explore SQuAD2.0 and model predictions
SQuAD2.0 paper (Rajpurkar & Jia et al. '18)

**SQuAD 1.1**, the previous version of the SQuAD dataset, contains 100,000+ question-answer pairs on 500+ articles.

## Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

| Rank | Model | EM | F1 |
|---|---|---|---|
| | Human Performance<br>*Stanford University*<br>(Rajpurkar & Jia et al. '18) | 86.831 | 89.452 |
| 1<br>Nov 06, 2019 | ALBERT + DAAF + Verifier (ensemble)<br>*PINGAN Omni-Sinitic* | **90.002** | **92.425** |
| 2<br>Sep 18, 2019 | ALBERT (ensemble model)<br>*Google Research & TTIC*<br>https://arxiv.org/abs/1909.11942 | 89.731 | 92.215 |
| 3<br>Jul 22, 2019 | XLNet + DAAF + Verifier (ensemble)<br>*PINGAN Omni-Sinitic* | 88.592 | 90.859 |
| 3<br>Sep 16, 2019 | ALBERT (single model)<br>*Google Research & TTIC*<br>https://arxiv.org/abs/1909.11942 | 88.107 | 90.902 |
| 3<br>Jul 26, 2019 | UPM (ensemble)<br>*Anonymous* | 88.231 | 90.713 |

# SQuAD(Stanford Question Answering Dataset)

```
{'data': [{'title': 'Super_Bowl_50',
'paragraphs': [{'context': 'Super Bowl 50 was an American football game to determine
the champion of the National Football League (NFL) for the 2015 season. The American
Football Conference (AFC) champion Denver Broncos defeated the National Football
Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title.
The game was played on February 7, 2016, at Levi\'s Stadium in the San Francisco Bay
Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized
the "golden anniversary" with various gold-themed initiatives, as well as temporarily
suspending the tradition of naming each Super Bowl game with Roman numerals (under
which the game would have been known as "Super Bowl L"), so that the logo could
prominently feature the Arabic numerals 50.',
'qas': [{'answers': [{'answer_start': 177, 'text': 'Denver Broncos'},
{'answer_start': 177, 'text': 'Denver Broncos'},
{'answer_start': 177, 'text': 'Denver Broncos'}],
'question': 'Which NFL team represented the AFC at Super Bowl 50?',
'id': '56be4db0acb8001400a502ec'}]}]}],
'version': '1.1'}
```

# SQuAD(Stanford Question Answering Dataset)

```
{"version": "v2.0",
"data": [{"title": "Normans",  "paragraphs": [{"qas": [{"question": "In what
country is Normandy located?",  "id": "56ddde6b9a695914005b9628",
"answers": [{"text": "France", "answer_start": 159}, {"text": "France",
"answer_start": 159}, "is_impossible": false}, {"plausible_answers": [{"text":
"10th century", "answer_start": 671}], "question": "When did the Frankish
identity emerge?", "id": "5ad39d53604f3c001a3fe8d4", "answers": [],
"is_impossible": true}], "context": "The Normans (Norman: Nourmands;
French: Normands; Latin: Normanni) were the people who in the 10th and
11th centuries gave their name to Normandy, a region in France. They were
descended from Norse (₩"Norman₩" comes from ₩"Norseman₩") raiders
and pirates from Denmark, Iceland and Norway who, under their leader Rollo,
agreed to swear fealty to King Charles III of West Francia. Through
generations of assimilation and mixing with the native Frankish and Roman-
Gaulish populations, their descendants would gradually merge with the
Carolingian-based cultures of West Francia. The distinct cultural and ethnic
identity of the Normans emerged initially in the first half of the 10th century,
and it continued to evolve over the succeeding centuries."}]
```

RAONBIT
DATA&SOLUTION

*AI Summit 2019 Seoul Workshop*

# 감사합니다

㈜라온비트

박진수

jinsu.park@raonbit.com