[neuron class] NLP

BERT-XDC/MRC

Day 1

Yongjae Lee
(yjlee@mindslab.ai, github.com/coalee)

MINDs Lab

NLP

Course Overview

지식 What → How

적용 Practice → Feedback

Brain XDC/MRC

= BERT-XDC/MRC

Bidirectional Encoder Representations from Transformers

Table of Contents: Day1

1. Representation Learning
2. BERT
3. XDC / MRC Engines & Data

참고자료:

1. 강희관 선임님 이전 강의자료들
2. Christopher Olah's blog (colah.github.io)
3. Jay Alammar's blog (jalammar.github.io)
4. several papers

0. Feel of NLP Task

마인즈랩(대표 유태준)은 클라우드 기반 회의록 자동정리 서비스 '마음회의록'의 성능 개선 작업과 기능 업그레이드로 인해 다양한 분야에서의 활용 가치가 높아졌다고 28일 밝혔다.

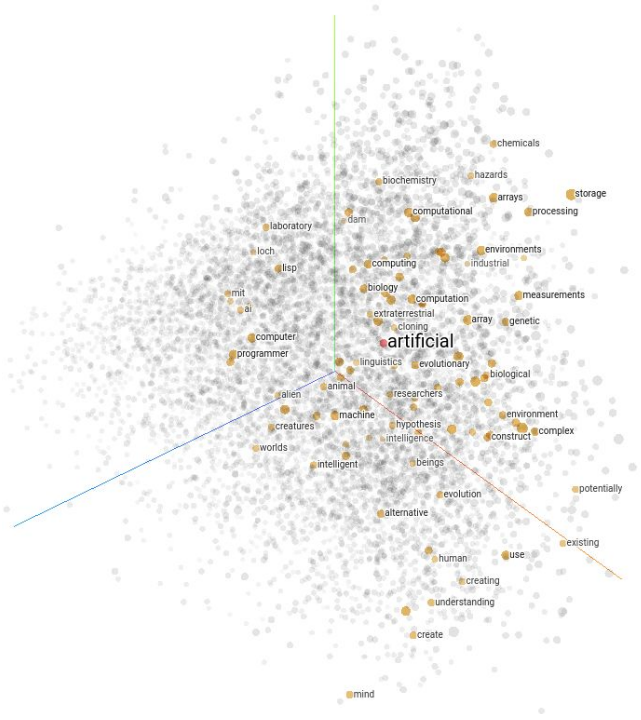
AI service firm MINDsLab expressed to be an official member of the Alberta Machine Intelligence Institute (Amii) and plan to proceed collaborative research with its researchers for three years starting April 1st 2019.

ברוב של 66 בעד ו-43 נגד: החוק הנורווגי אושר
הלילה (בין שני לשלישי) בכנסת. הדיון על התיקון
שמאפשר לח"כים חדשים לעבור למפלגה אחרת תוך
24 שעות הסתיים - והחוק עבר באופן סופי על כלל
סעיפיו. במקביל, העימותים בין כחול לבן לליכוד נמשכו
כל היום וגנץ הודיע על כינוס "שיחת עדכון" לסיעתו.
מוקדם יותר פורסם במהדורה המרכזית כי ראש
הממשלה הבהיר בשיחות סגורות שהוא "הולך על
הסיפוח בכל הכוח", במפלגתו של גנץ ענו על הפרסום:
"אין לנו עמדה כי נתניהו לא הציג לנו שום מפה".
במקביל לכל אלה, בהצבעה הראשונה בכנסת על
החוק הנורווגי אנשי נתניהו אמרו שהוא לא יגיע בגלל
שיש רוב גם בלעדיו.

??

(from Google News)

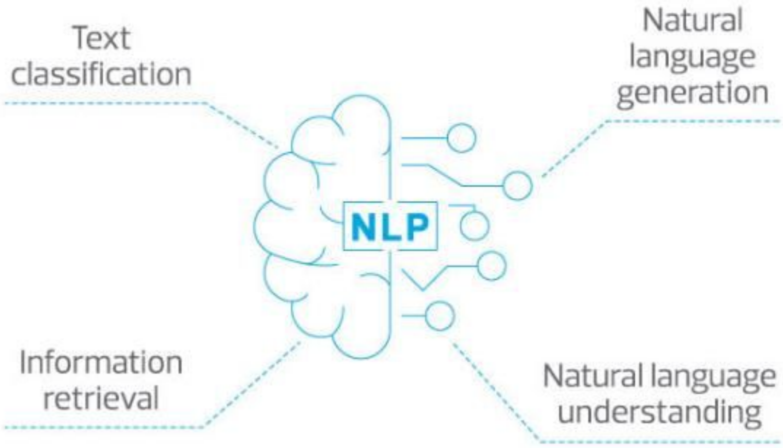
Representation Learning



I. Representation Learning

NLP: Natural Language Processing

- NL Understanding
- NL Generation



자연어 Natural Language

- 인간의 정보 전달 수단
- 인간 고유의 능력
- 인공 언어에 대응되는 개념
- 특정 집단에서 사용되는 모국어 집합

처리 Processing

- **How to represent** NL in/for processor(CPU or GPU)?

I. Representation Learning

MINDs Lab

How to represent natural language? Naive approach

Numbering

- indexing from 0 to n
- cheap(1-dim), meaningless

One-hot vector

- 특정 차원만 1, 나머지는 0인 vector 표현
- very expensive(n-dim), still meaningless

자연어	자연어처리는 재밌다
Tokenized	[자연, 어, 처리, 는, 재미, ㅆ, 다]
Numbering	[43, 563, 293, 3, 1022, 57, 4]
One-hot	[[0, 1, 0, …, 0], … [0, …, 0, 1, 0, …, 0]]
Word Embedding	[[0.1, 0.73, -0.34, ...], … [-0.6, 0.22, 0.12, ...]]

I. Representation Learning

How to represent natural language? Word Embedding (1/2)

Embedding: 고차원(n-dim) 데이터 → 저차원(k-dim)상의 연속성 있는 표현

- 공간 상에서 의미를 가질 수 있다
- 유사한 데이터 군집화(clustering)

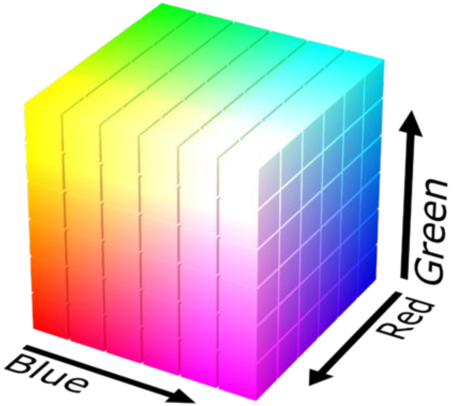
ex) RGB: 3차원(256 * 256 * 256)으로 색 표현.

Pastel Green(#77DD76)

Conditioner(#FCFFC6)

Blizzard Blue(#BBE3F1)

#FF0033, #111111, #EEEEEE



wikimedia

I. Representation Learning

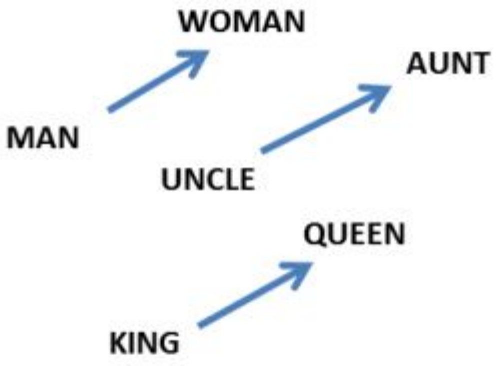
How to represent natural language? Word Embedding (2/2)

Word Embedding

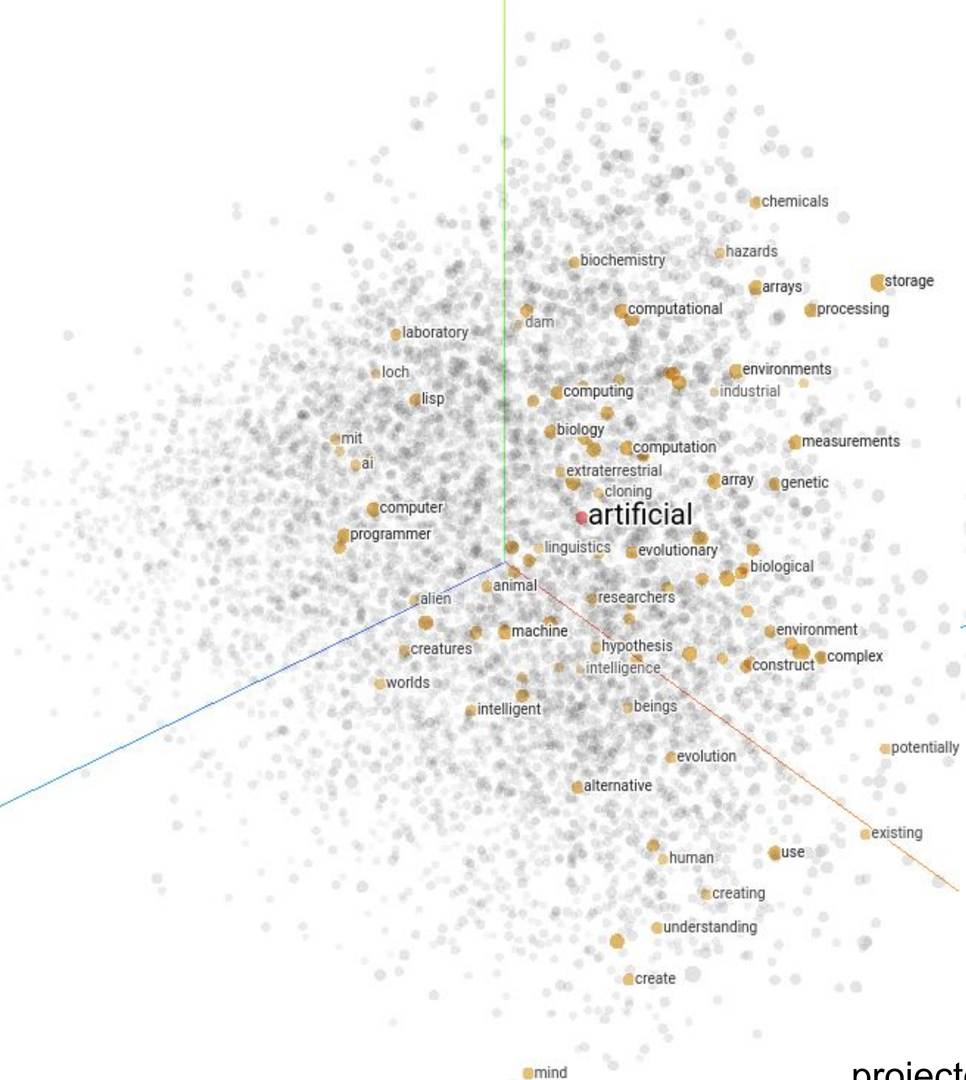
- 대량의 문서를 이용해 각 단어를 벡터 표현으로 **학습**
- Algorithms: CBoW, Skip-gram, GLoVe, etc.
- 학습된 이후에는 유사도 측정, 벡터 연산 등을 수행할 수 있

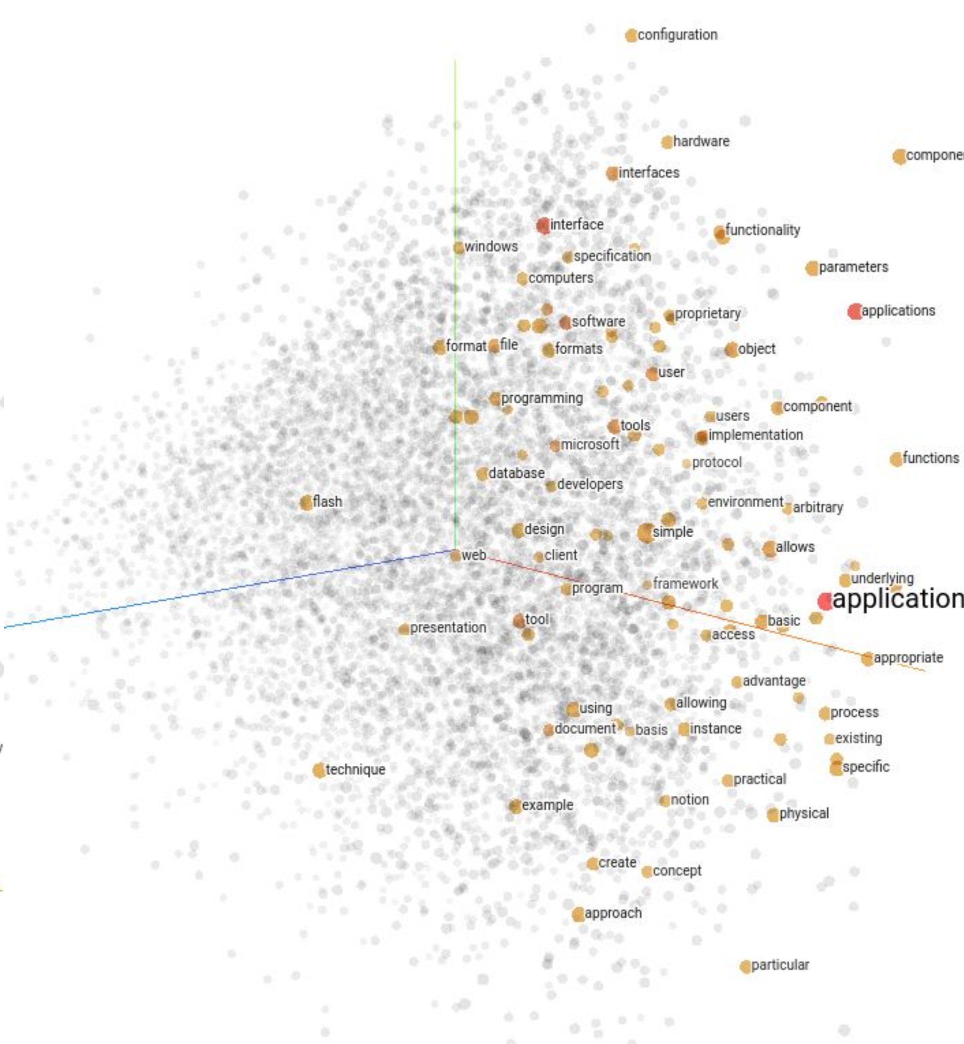
- w('한국') - w('서울') + w('도쿄') → w('일본')
- w('회사') + w('인공지능') → w('벤처기업')

word2vec.kr projector.tensorflow.org

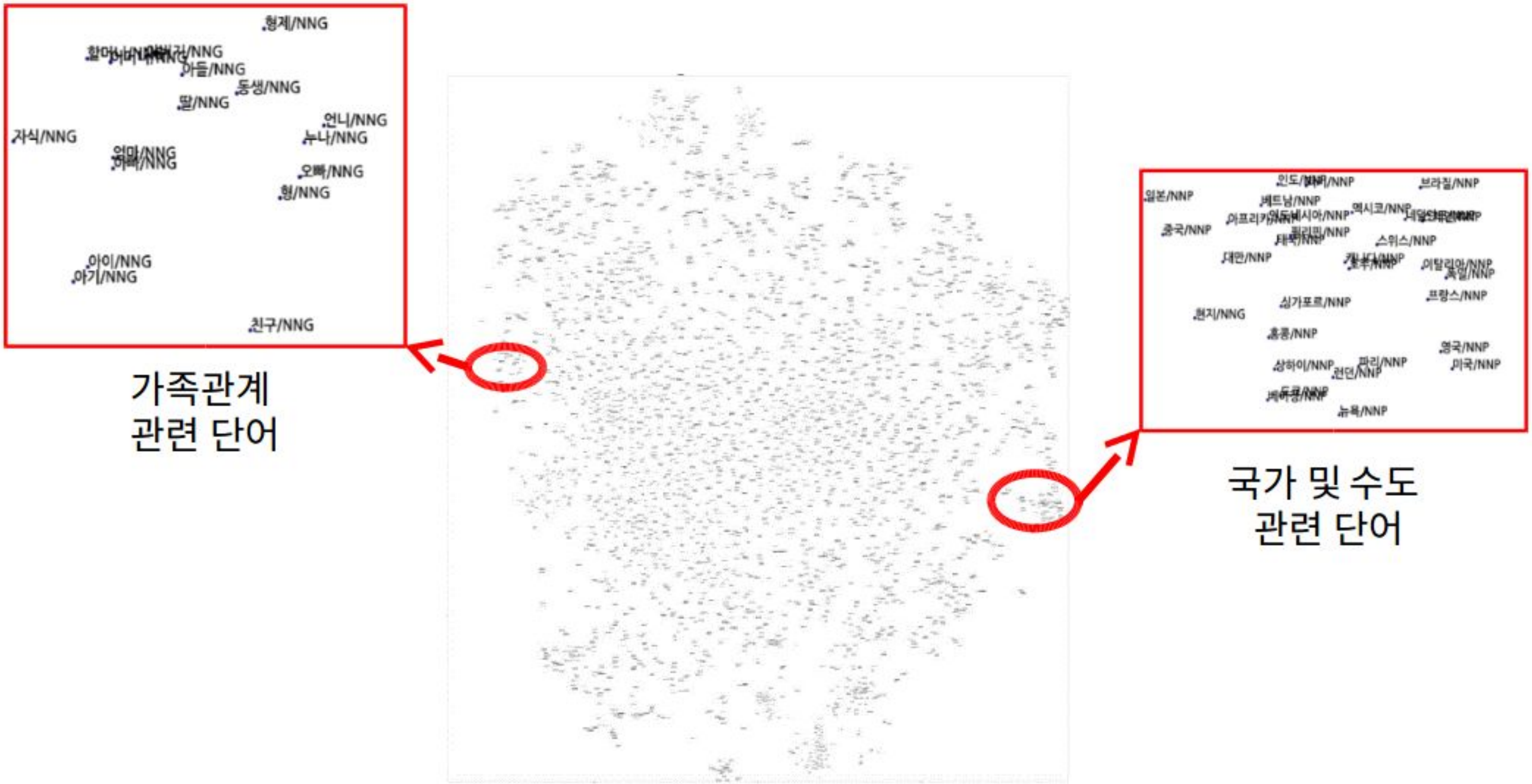


Mikolov, et al. (2013a)





projector.tensorflow.org



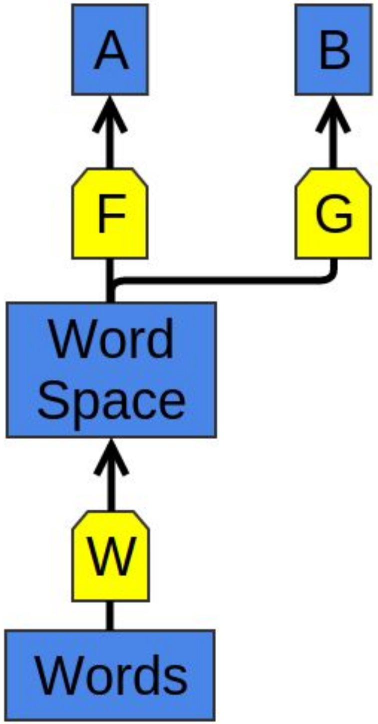
Word Embedding Visualization

I. Representation Learning

Then, **how to learn** a (good) representation? (1/2)

The use of word representations… has become a key ‘secret sauce’ for the success of many NLP systems in recent years, across tasks including named entity recognition, part-of-speech tagging, parsing, and semantic role labeling.
(Luong, et al. 2013)

This general tactic – **learning a good representation on a task A and then using it on a task B** – is one of the major tricks in the Deep Learning toolbox. It goes by different names depending on the details: pretraining, transfer learning, and multi-task learning.
(Christopher Olah’s post, 2014. emphasis added)

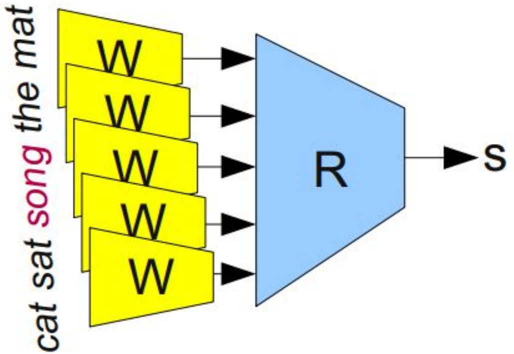


Christopher Olah, 2014

I. Representation Learning

MINDs Lab

Then, **how to learn** a (good) representation? (2/2)



Bottou, 2011

ex) Bottou, 2011

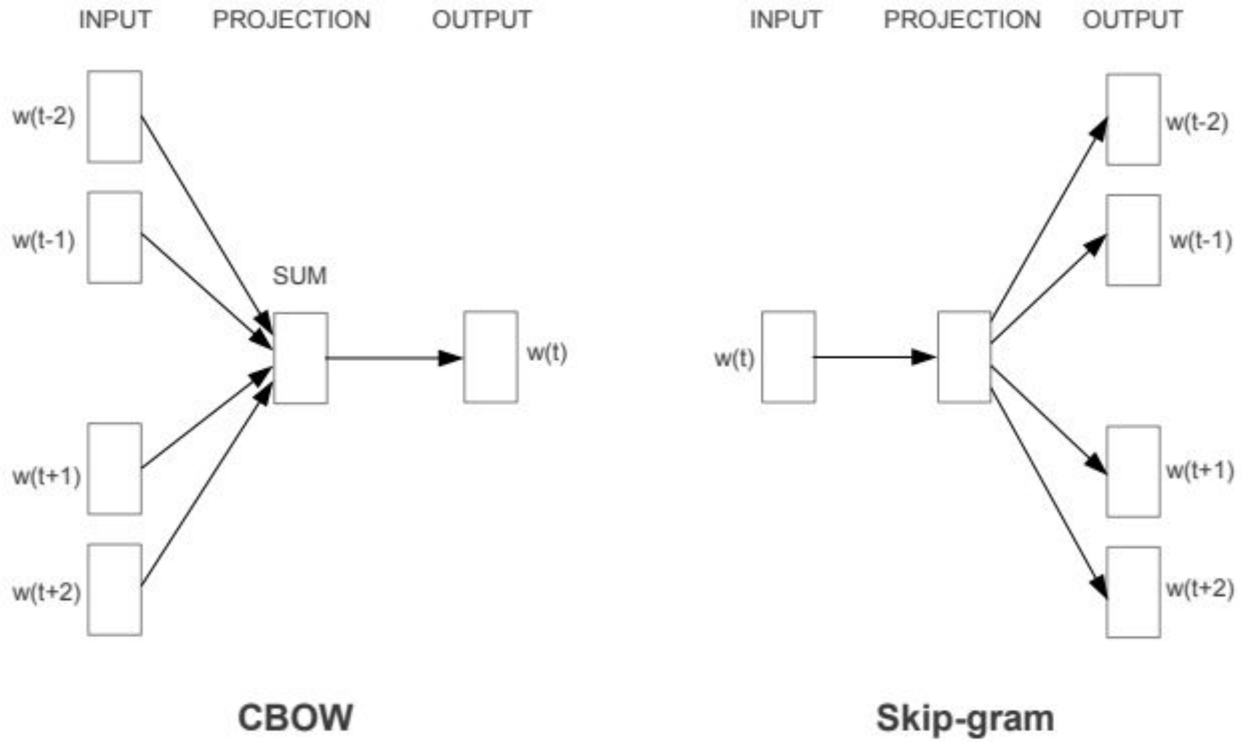
1. 문장 추출 from large text corpora(wikipedia)
   ex) *cat sat on the mat*
2. *W* (word embedding) 랜덤 초기화
3. *R* (5-gram validity module) 학습

valid: *R*(*W*("cat"), *W*("sat"), *W*("on"), *W*("the"), *W*("mat"))=1

invalid: *R*(*W*("cat"), *W*("sat"), *W*("song"), *W*("the"), *W*("mat"))=0

more ideas: CBoW, Skip-gram, GLoVe, ELMo, BERT

CBoW, Skip-gram (Mikolov, 2013)



I. Representation Learning

ELMo: **E**mbeddings from **L**anguage **Mo**del (Peters, et al. 2018.02)

- 기존 방식의 한계: 고정된 벡터로는 문맥을 반영하지 못함
- 문맥을 반영한 워드 임베딩 **ELMo**의 등장

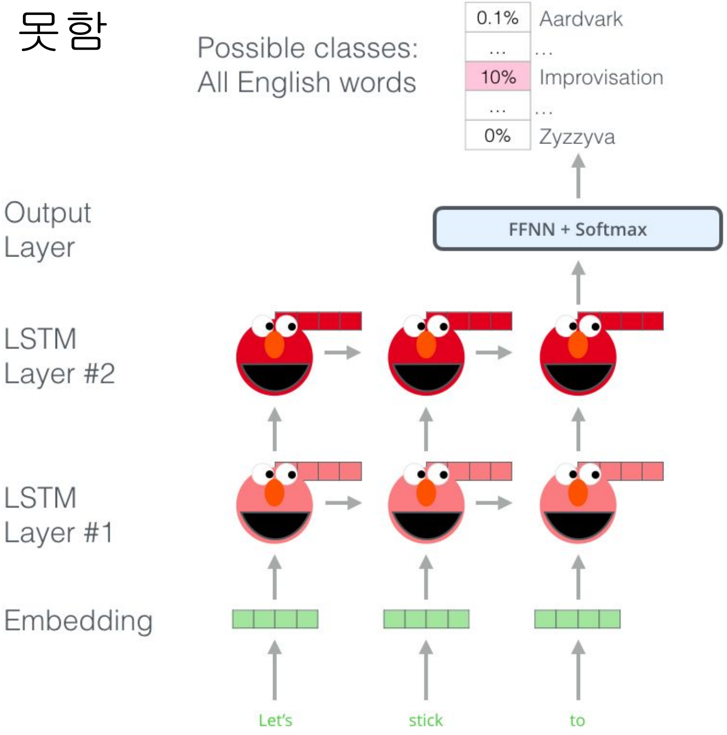
ex) 'bank account', 'river bank'에서 bank는 다른 의미

close a <u>bank</u> account → bank: [-0.2, 0.3, …]

walk along a river <u>bank</u> → bank: [0.4, -0.1, …]

Language Model:

- predicts a(next) token based on other tokens
- 대량의 텍스트로 un(semi-)supervised 학습



Jay Alammar

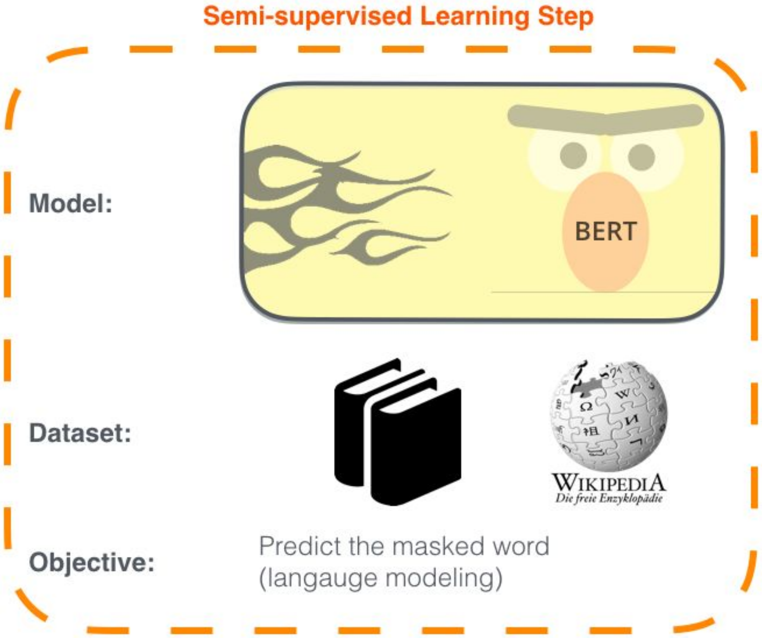
I. Representation Learning

MINDs Lab

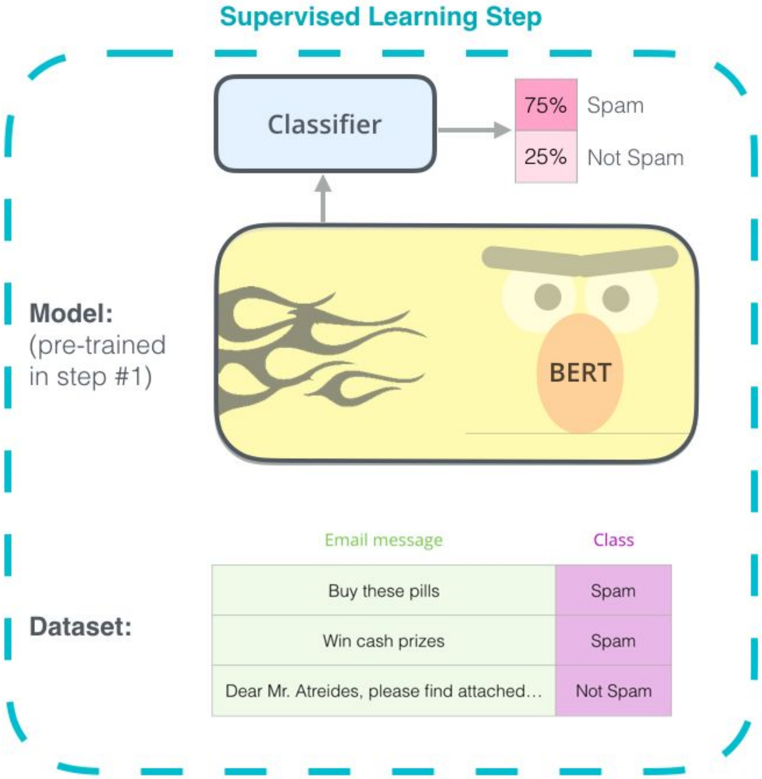
BERT

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.



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



Quiz 1. Representation Learning

1. Why do we need Representation Learning?

2. What's the difference between two ways of word representation:
    - one hot vector
    - word embedding

BERT;

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

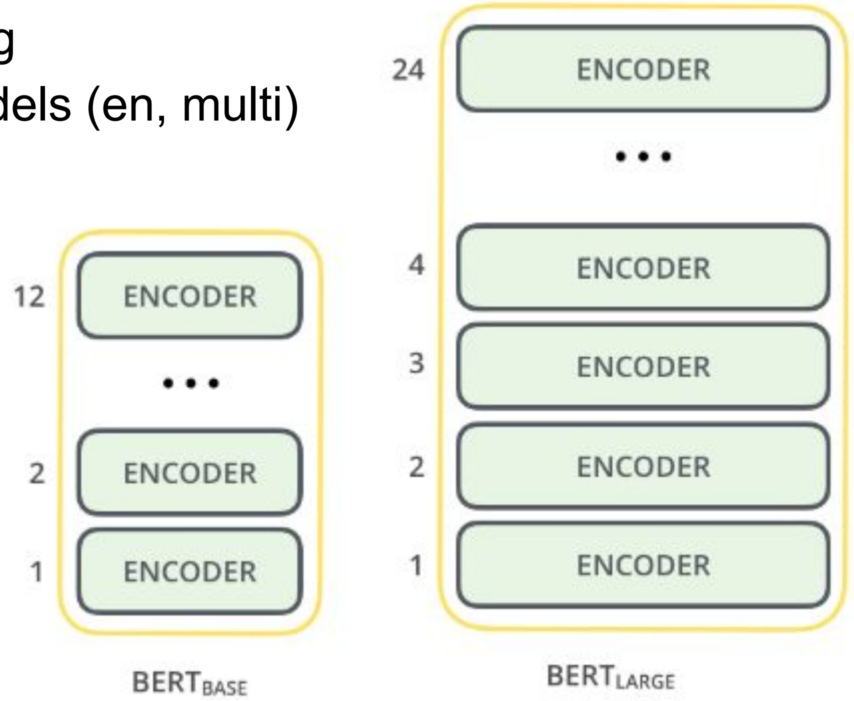
II. BERT: Bidirectional Encoder Representations from Transformers

MINDs Lab

BERT

- Utilizes Transformers' Encoder block
- Transfer learning: pre-training → fine-tuning
- Google opened the code & pre-trained models (en, multi)
  → it boosts NLP !

models:
BERT Base: L12_H768_A12 → 110M
BERT Large: L24_H1024_A16 → 330M
+ BERT Tiny, Mini, Small, Medium (L2~8)



II. BERT: Bidirectional Encoder Representations from **Transformers**

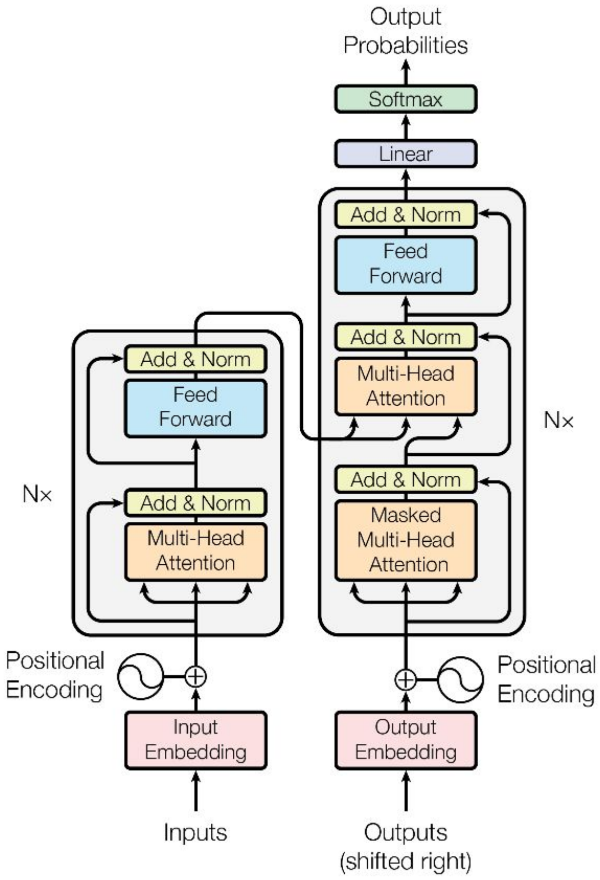
MINDs Lab

Transformers. *Attention is All You Need*. (Vaswani, et al. 2017.06)

- seq2seq architecture
- Neural Machine Translation
- Utilizes attention layers only
  self-attention: RNN, CNN의 단점 극복
  usual(aligning) attention

- RNN, CNN 단점: 연산량, long-term dependency

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



(Vaswani, et al, 2017)

Attention (NMT by Jointly Learning to Align and Translate, Cho et al. 2014.09)

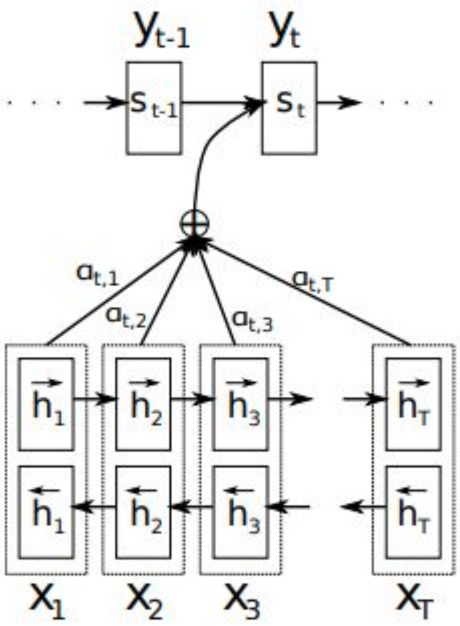
- Query, Key, Value
- Query on {Key:Value}

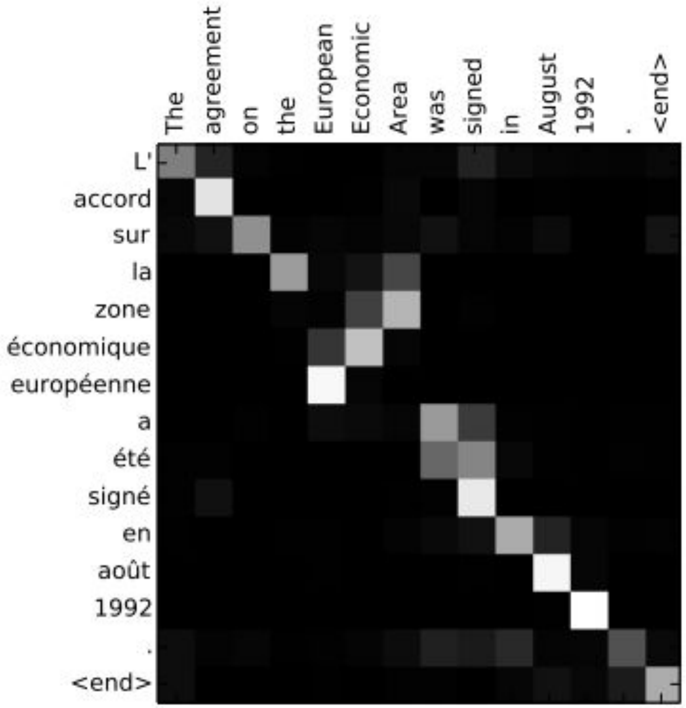
- Learn to **align**

  (→ Explainability !)

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$





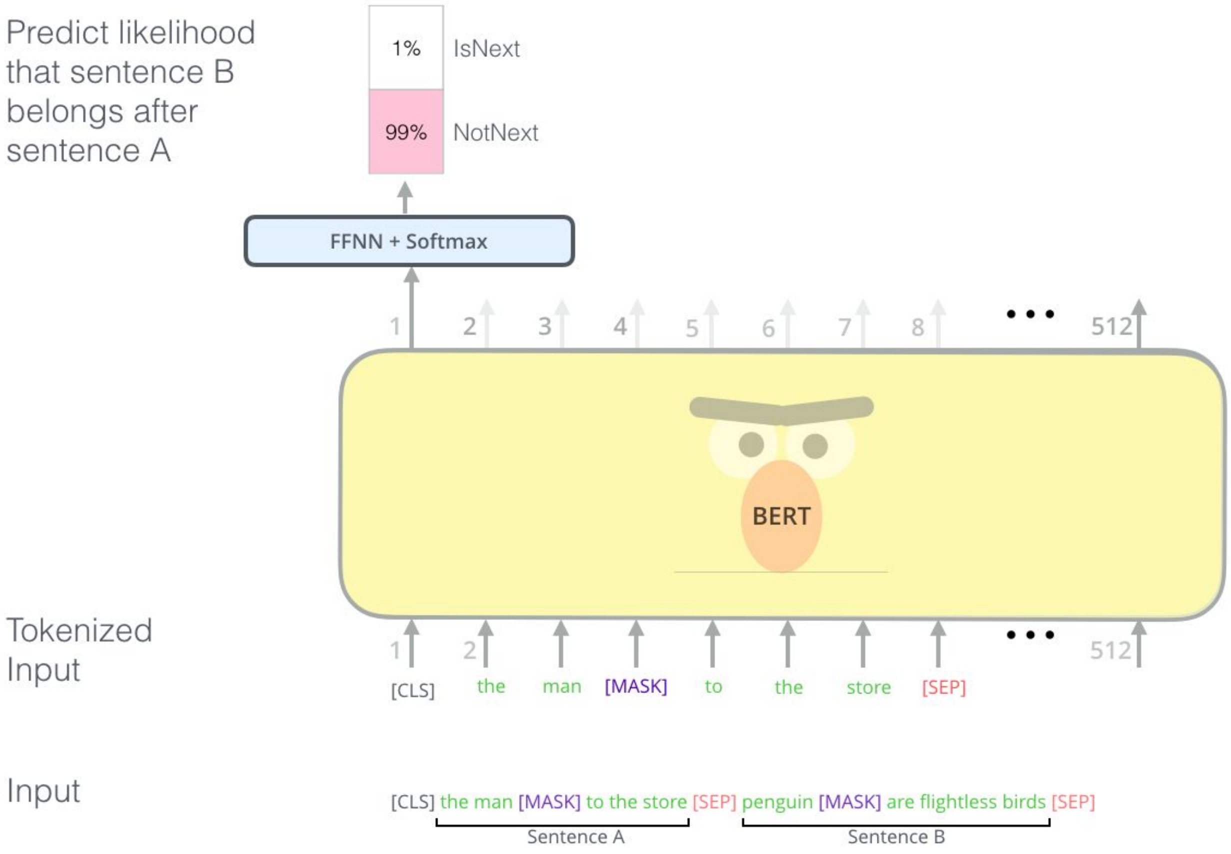
(Bahdanau, Cho, Benzio, 2014)

II. BERT: Bidirectional Encoder Representations from Transformers

Pre-training Tasks

1. **MLM**:
Masked LM

2. **NSP:**
Next Sentence Predict



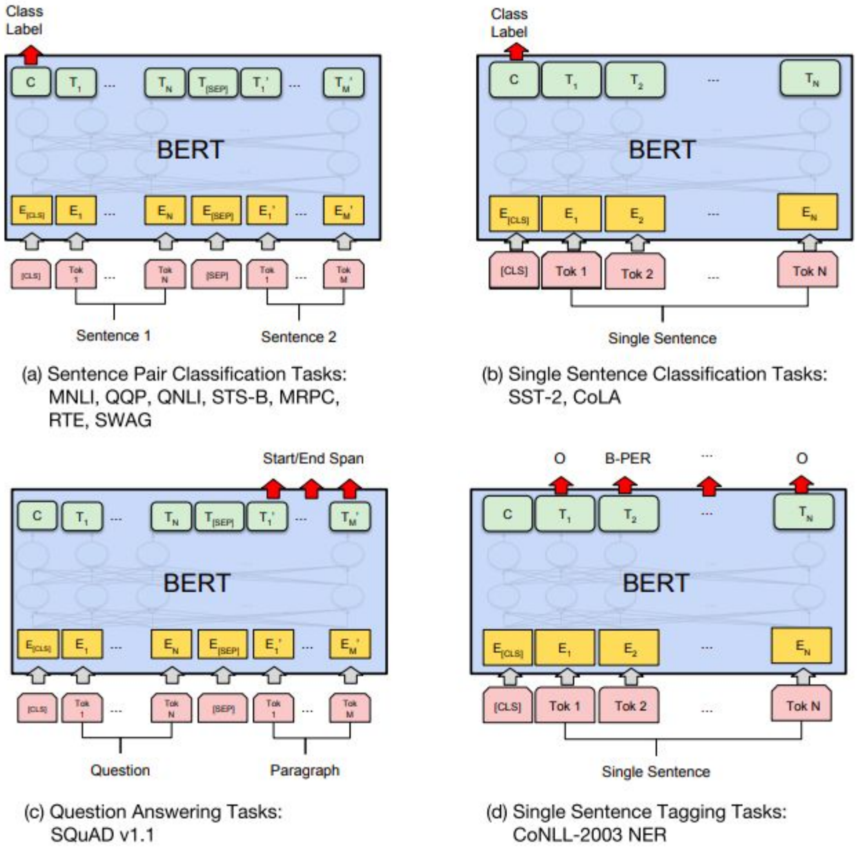
II. BERT: Bidirectional Encoder Representations from Transformers

BERT Fine-tuning

1. Load pre-trained model

2. Input:
한 문장 혹은 두 문장([SEP]로 구분)

3. Output:
Classification: class label
Answering: begin & end index
Entity: tag



(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

Quiz 2. BERT

1. What are the two pre-training tasks of BERT?

2. What is the name of the process after pre-training, and how it goes?

3. What is the advantage of attention mechanism?

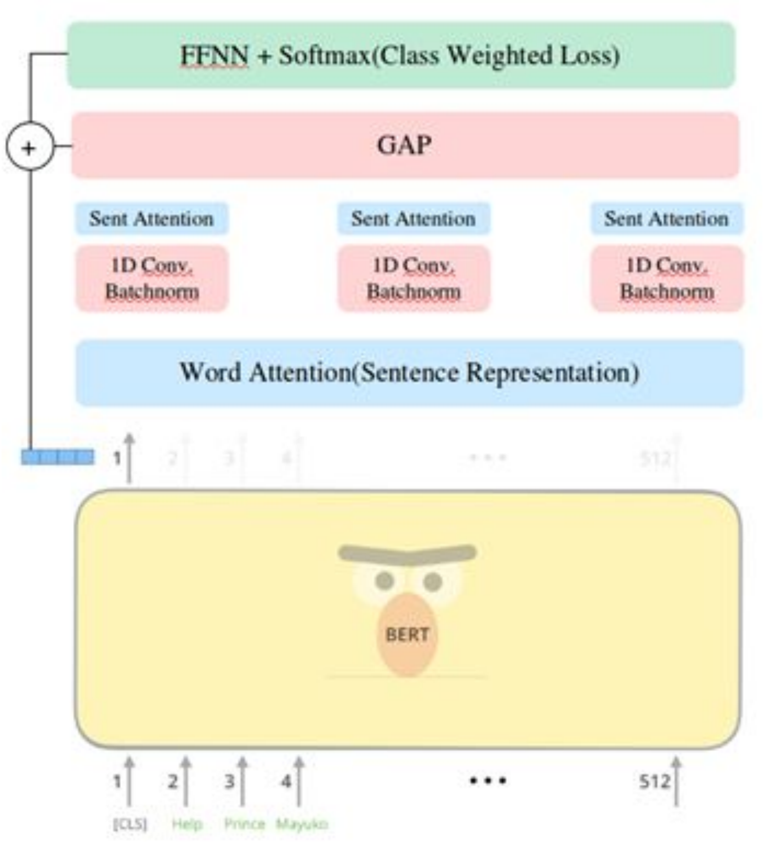
XDC / MRC
Engines & Data

III. XDC / MRC

XDC: eXplainable Document Classifier

BERT-XDC = BERT + Attention + Classifier

Input: passage (tokenized seq length <= 512)
output: label, attentions(sentence, word)

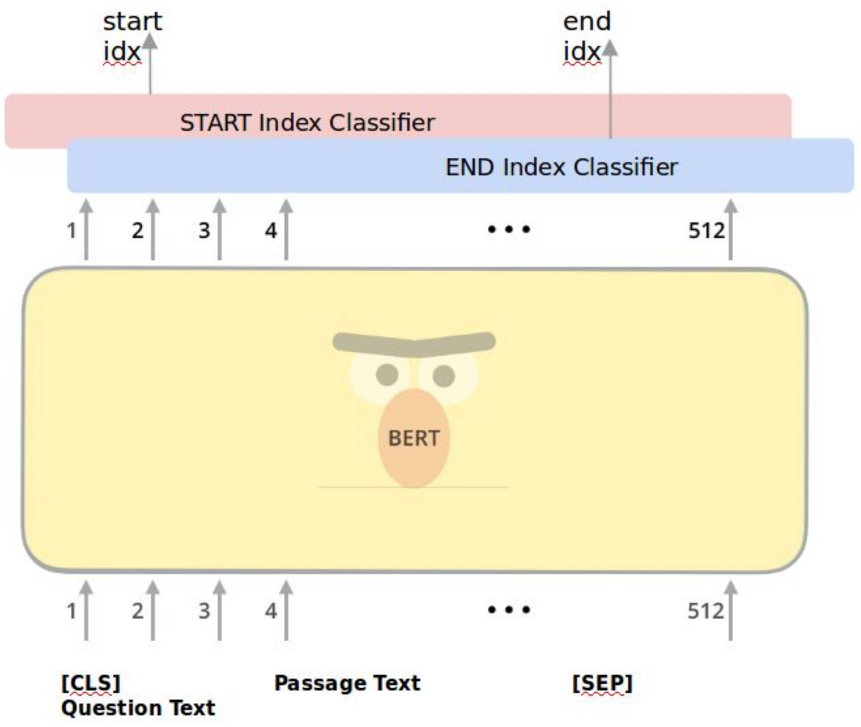


III. XDC / MRC

MRC: Machine Reading Comprehension

BERT-MRC = BERT + MRC(QnA)

Input: context, question (length <= 512)
output: start & end index on context



III. XDC / MRC

XDC 학습 및 실행 Process

- 기본적으로 아래와 같이 3 단계로 진행하며, 성능 안정화가 될 때까지 반복 작업
- 사전학습은 다량의 데이터와 오랜 시간이 필요하기 때문에 미리 학습된 모델을 활용

데이터 수집 (Data Gathering)

✓크롤링 등의 방법으로 대량의 Text 데이터 수집

✓전처리 : Data format 통일화

XDC 학습 (XDC Train)

✓학습데이터 Tokenization

✓Pre-trained model 기반 학습 진행

실행 및 테스트 (XDC Inference)

✓Input Data Tokenization

✓다양한 인자값을 조절하며 학습 결과 확인

III. XDC / MRC

BERT MRC 학습 및 실행 Process

- 기본적으로 아래와 같이 3 단계로 진행하며, 성능 안정화가 될 때까지 반복 작업

데이터 수집 (Data Gathering)

✓크롤링 등의 방법으로 대량의 Text 데이터 수집
✓Data format 통일화

BERT MRC 학습 (BERT MRC Train)

✓학습데이터 Tokenization
✓Pre-trained model 기반 학습 진행

실행 및 테스트 (BERT MRC Inference)

✓Input Data Tokenization
✓다양한 인자값을 조절하며 학습 결과 확인

III. XDC / MRC

MINDs Lab

데이터 규격

	Train (/ Test) data	Inference input
XDC	train_data.txt (*context* + ‘\t’ + *label* + ‘\n’) x N	python xdc_inference.py --context *context*

MRC	train_data.json (KorQuAD 규격)
	{‘paragraphs’: [{
	‘context’: *context,*
	‘qas’: [{
	‘answers’: [{
	‘answer_start’: *start_idx*,
	‘text’: answer_*text,*
	‘id’: *id*,
	‘question’: *question_text*
	}, …]
	}, …]
	}, ...]}
		python mrc_inference.py
		--context *context*
		--question *question*

Quiz 3. XDC / MRC

1. Components of BERT-XDC engines?
2. Input / output of XDC?
3. Components of BERT-MRC engines?
4. Input / output of MRC?

Assignment

XDC / MRC 실습을 위한 데이터 준비

XDC 데이터

- 선택: 제공된 데이터 가공 | 뉴스 데이터 크롤링 | 원하는 데이터 크롤링

- 되도록 10K(최소 2K~) 이상, 데이터 규격에 맞춰서 파일 업로드 (neuron 팀 안내)
- 뉴스 데이터는 테스트 데이터로 성능 측정
  test labels: ['문화', '정치', '미용/건강', '생활', 'IT/과학', '사회', '경제', '스포츠', '연예']

MRC 데이터

- KorQuAD 1.0 Open Dataset (train/dev)

Paper references

Bottou. From Machine Learning to Machine Reasoning, 2011

Mikolov, et al. Efficient Estimation of Word Representations in Vector Space, 2013

Mikolov, et al. Distributed Representations of Words and Phrases and their Compositionality, 2013

Bahdanau, et al. Neural Machine Translation by Jointly Learning to Align and Translate, 2014

Vaswani, et al. Attention Is All You Need, 2017

Peters, et al. Deep contextualized word representations, 2018

Devlin, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

감사합니다

MINDs Lab