KOREA UNIVERSITY

DSBA CS224n 2021 Study

# [Special Lecture 2]
# What Does BERT Look At? An Analysis of BERT's Attention

DSBA Data Science & Business Analytics

고려대학교 산업경영공학과

Data Science & Business Analytics Lab

발표자 : 김도윤

# 00 발표 목차

Contents

# 1 Introduction

# 01 Introduction

What does BERT look at?

- Pre-trained language model의 극적인 Fine-tuning 효과

- 도대체 why?
  - ✓ 모델이 사전학습을 통해 언어 구조(Language Structure)를 파악함

- 어떠한 언어적 특징을 학습하는 것인지 확인하는 방법은?
  - ✓ 선별하여 입력한 문장과 결과를 통해 확인
  - ✓ 중간 산출물인 vector representation 으로 probing classifier 를 평가

- **본 논문에서는 사전 학습된 BERT 모델의 Attention maps을 통해 각 attention head의 특성을 확인함**

# 01 Introduction

## Main points

1. 물리적인 패턴을 확인 : How BERT's attention heads behave
   - ✓ 특정 위치로 혹은 넓게 attending 하는 패턴 등을 발견
   - ✓ [SEP]에 상당수 집중함, no-op 관계를 의미
   - ✓ 같은 layer에 있는 attention head들의 유사한 behavior

2. Attention head 가 어떠한 언어적 현상을 파악하는지 조사함 : Probe each attention head for linguistic phenomena
   - ✓ Attention head를 하나의 모델로 간주하여 입력된 단어의 attention score를 가지고 단어들 사이의 syntactic relation 잘 파악되는지 확인
   - ✓ 특정 attention head 가 특정 syntactic relation을 잘 찾아내는(예측하는) 현상을 발견

     Ex) 관사-명사, 타동사-직접 목적어, 전치사-전치사의 목적어, 소유격 대명사 등
   - ✓ Coreference resolution 에 대해서도 잘 파악되는지 확인
   - ✓ 이러한 attention head의 behavior는 self-supervised training에 의한 것으로 판단

3. Attention-based probing classifier 제안
   - ✓ Dependency parsing 에 대한 예측력 확인

## 2 Background: Transformers and BERT

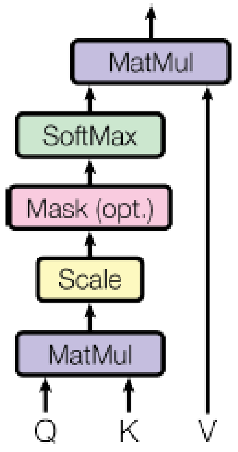
# 02 Background

Transformer [Attention is all you need (A Vaswani et al., 2017)]

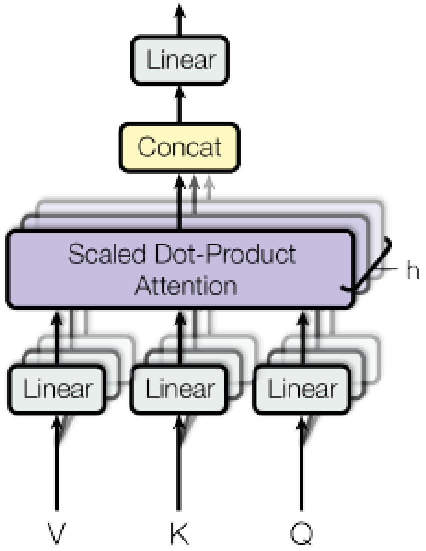
김동화 박사 Transformer & BERT 설명 영상 [https://youtu.be/xhY7m8QVKjo](https://youtu.be/xhY7m8QVKjo)

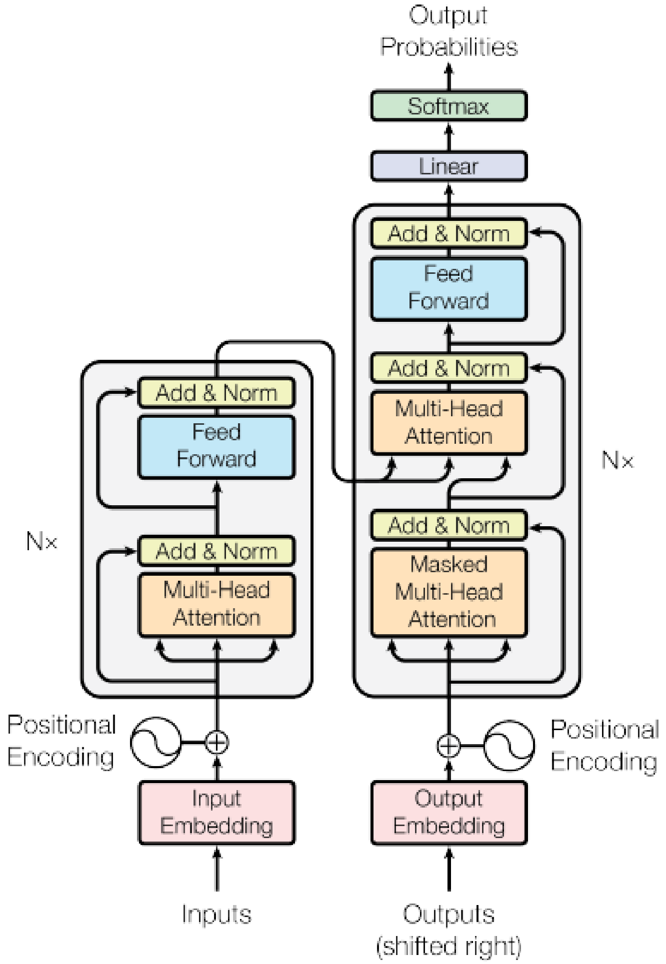
- $Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
- $MultiHead(Q,K,V) = Concat(head_1, head_2, \dots, head_h)W^O \in R^{d_{model}}$
  - ✓ $h = 8\,,\ d_{model} = 512,\ \rightarrow d_k = d_v = d_{model}/h = 64$
  - ✓ $Q,K,V \in R^{d_{model}} \rightarrow \boxed{QW_i^Q, KW_i^K \in R^{d_k},\ \ VW_i^V \in R^{d_v}}$
  - ✓ $W_i^Q \in R^{d_{model}\times d_k},\ W_i^K \in R^{d_{model}\times d_k},\ W_i^V \in R^{d_{model}\times d_V}, W^O \in R^{hd_v \times d_{model}}$
  - ✓ $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \in R^{d_v}$

Scaled Dot-Product Attention




Multi-Head Attention




Transformer 구조

# 02 Background

BERT [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding(Devlin et al., 2019)]

김동화 박사 Transformer & BERT 설명 영상 https://youtu.be/xhY7m8QVKjo

- A multi-layer bidirectional Transformer encoder

- Model Setting
  - ✓ L = The number of layers (Transformer Encoder block)
  - ✓ H = Hidden size
  - ✓ A = The number of self-attention heads

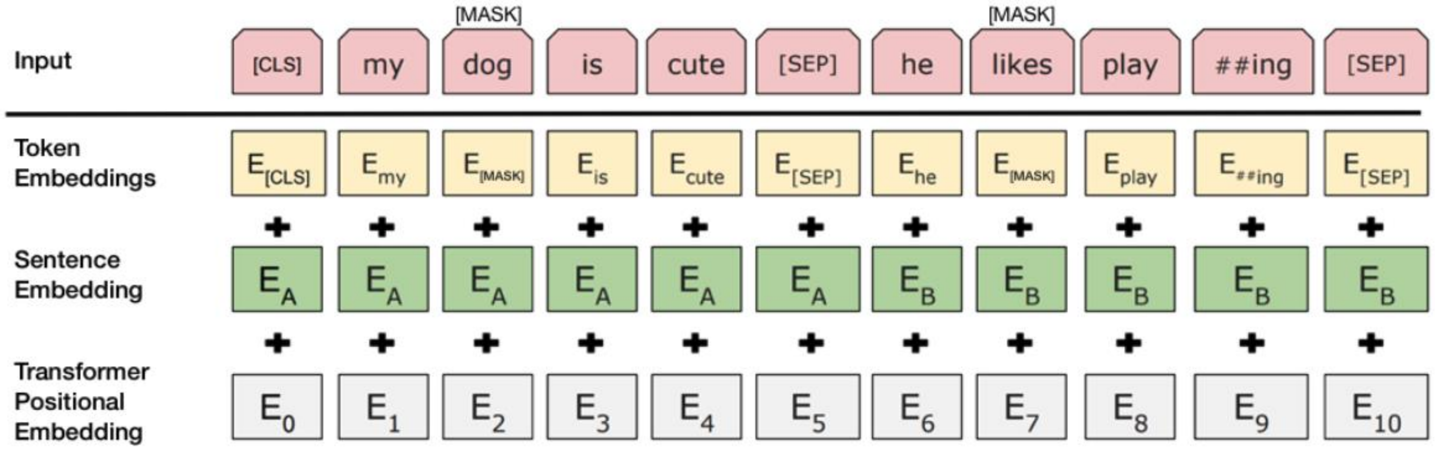
- Model Type
  - ✓ <u>BERT-base : [L=12, H=768, A=12, Total Parameters = 110M]</u>
  - ✓ BERT-Large : [L=24, H=1024, A=16, Total Parameters = 340M]

*Notation
<layer>-<Head>

Ex) Head 8-7
= 8번째 layer에 있는 7번째 Head

- Pretraining Tasks
  - ✓ Masked Language Model
  - ✓ Next Sentence Prediction

- Preprocessing using special Tokens
  - ✓ [CLS], [SEP]



## 3 Surface-Level Patterns in Attention

03

# Surface-Level Patterns in Attention

## Setup

- BERT-base 모델을 1,000개의 Wikipedia paragraph segment 를 입력
- 하나의 paragraph segment 당 128개의 토큰 포함
- Input : [CLS]-<paragraph1>-[SEP]-<paragraph2>-[SEP]
- 입력 시 masking은 안 함 : Attention 의 온전한 behavior를 보기 위함
- Model의 configuration은 BERT-base를 따라감 [L = 12, H = 768, A = 12, Batch size = 16]
- Tensors
  - ✓ Total Number of heads : 12 layers x 12 heads = 144 heads
  - ✓ $A = 12\,,\ d_{model} = 768,\ \rightarrow d_k = d_v = d_{model}/h = 64$
  - ✓ $Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \in R^{64}$
  - ✓ $head_{<l-i>} = Attention_l\left(QW_i^Q, KW_i^K, VW_i^V\right) \in R^{64}$ , l = 1, …, 12 , i = 1, …, 12
  - ✓ $Q,K,V \in R^{768} \rightarrow QW_i^Q, KW_i^K \in R^{64},\ \ VW_i^V \in R^{64}$
  - ✓ $W_i^Q \in R^{768\times 64},\ W_i^K \in R^{768\times 64},\ W_i^V \in R^{768\times 64}, W^O \in R^{768\times 768}$

# Surface-Level Patterns in Attention

## Setup

https://github.com/clarkkev/attention-analysis/blob/master/General_Analysis.ipynb

- Sequence
  - ✓ $\mathrm{h} = [[CLS], \mathrm{w}_{1,1}, \mathrm{w}_{1,2}, \ldots \mathrm{w}_{1,128}, [\mathrm{SEP}], \mathrm{w}_{2,1}, \mathrm{w}_{2,2}, \ldots \mathrm{w}_{2,128}, [\mathrm{SEP}]]$

- Head 별 Attention 구하는 방법

```
avg_attns = {
    k: np.zeros((12, 12)) for k in [
      "self", "right", "left", "sep", "sep_sep", "rest_sep",
      "cls", "punct"]
}
```

```
# get the average attention for each token type
for key, selector in selectors.items():
  if key == "sep_sep":
    denom = 2
  elif key == "rest_sep":
    denom = n_tokens - 2
  else:
    denom = n_tokens
  avg_attns[key] += (
      (attns * selector[np.newaxis, np.newaxis]).sum(-1).sum(-1) /
      (n_docs * denom))
```

avg_attn[key].shape=(12,12)

# Surface-Level Patterns in Attention

Relative Position

https://github.com/clarkkev/attention-analysis/blob/master/General_Analysis.ipynb

- Most Heads put little attention on the current token(=자기 자신)
- Specialized heads attending heavily on the 'next' or 'previous' token in <u>early layers</u>
  - ✓ In layer 2, 4, 7, 8 : on average put over 50% of their attention on the previous token
  - ✓ In layer 1, 2, 2, 3, 6 : on average put over 50% of their attention on the next token

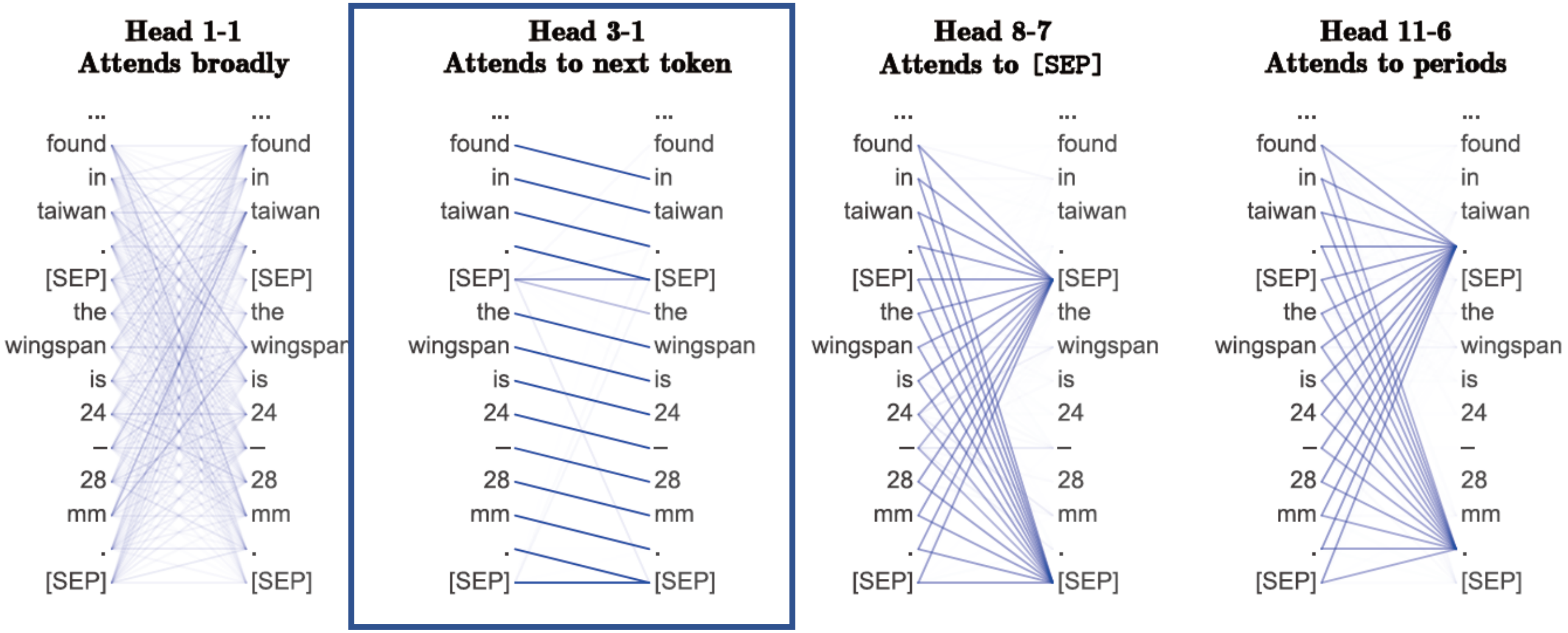



Figure 1: Examples of heads exhibiting the patterns discussed in Section 3. The darkness of a line indicates the strength of the attention weight (some attention weights are so low they are invisible).

# 03 Surface-Level Patterns in Attention

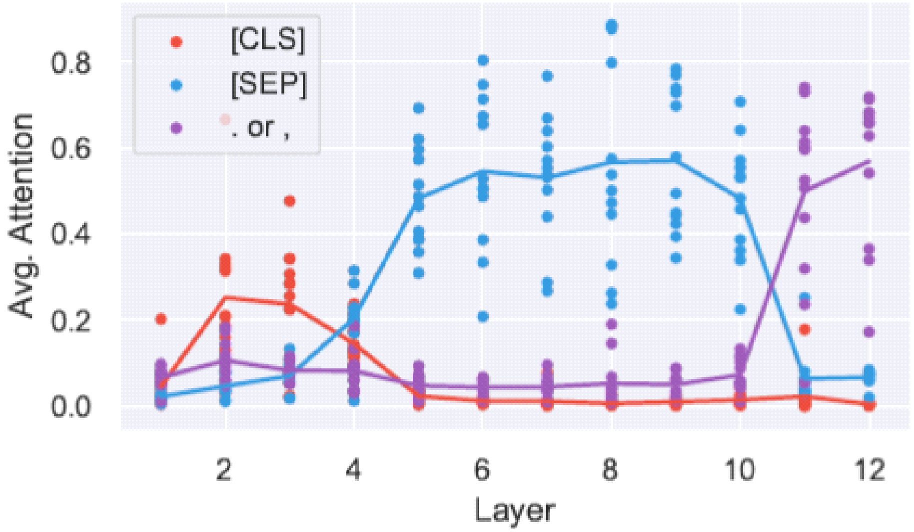
## Attending to Separator Tokens

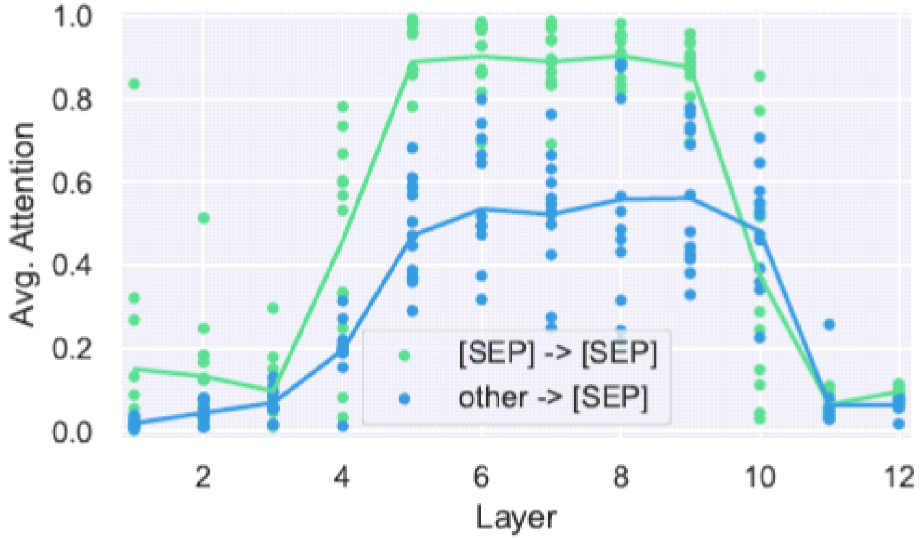
https://github.com/clarkkev/attention-analysis/blob/master/General_Analysis.ipynb

- 상당수 [SEP] Token으로 Focus 됨
- 좌측 그림 : 해당 token으로 향한 attention 평균 값 / 우측 그림 : [SEP] to [SEP], 다른 토큰 to [SEP]의 attention 평균 값
- Layer 6-10 : 절반이 넘게 [SEP] 으로 Attention이 집중됨
  - ✓ [SEP]이 2번씩 등장 하는 이유 때문?!

*"since most of our segments are 128 tokens long, the average attention for a token occurring twice in a segments like [SEP] would normally be around 1/64."*

- [SEP]가 segment-level에서 segment를 압축하는 역할을 할까?
  - ✓ 그것은 아님! If so, [SEP]이 다른 token들로 broadly attending 해야 하는데 주로 자기 자신 혹은 다른 [SEP]로 몰림



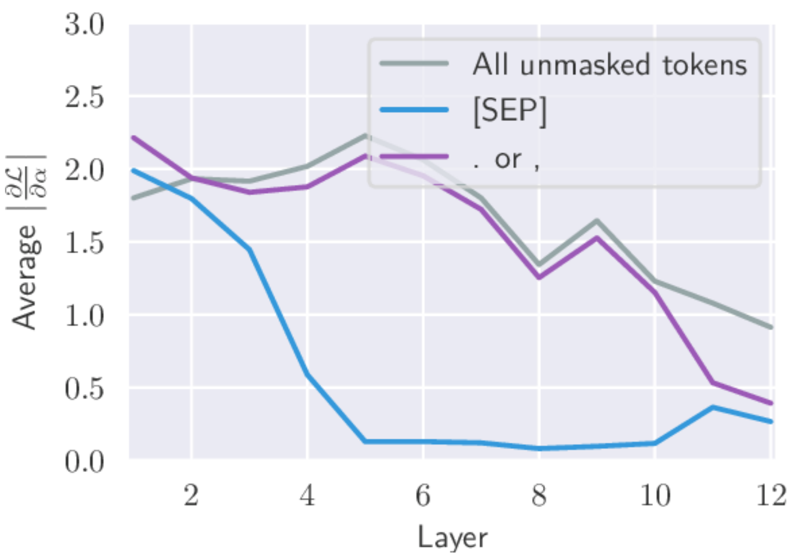
# 03 Surface-Level Patterns in Attention

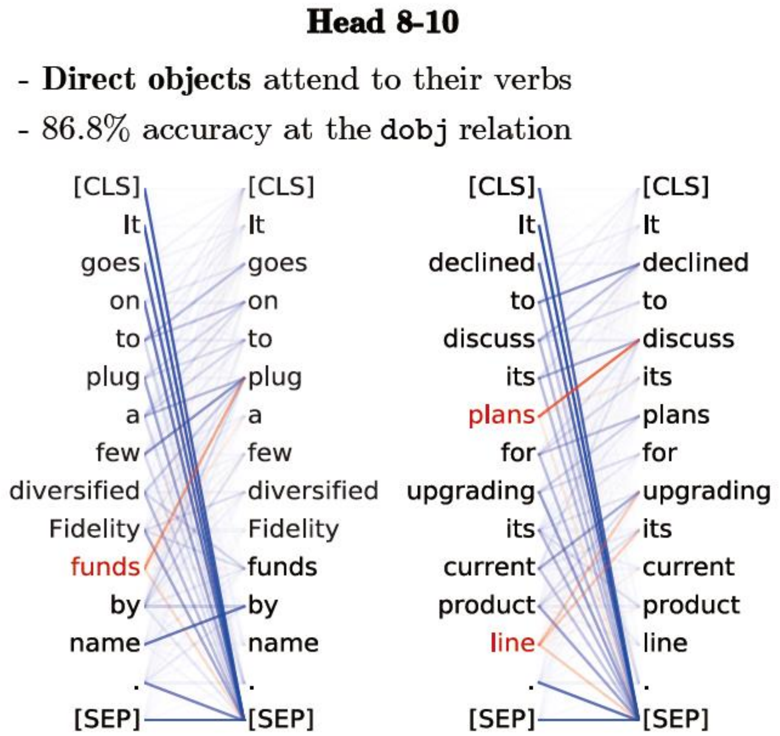
Attending to Separator Tokens

https://github.com/clarkkev/attention-analysis/blob/master/General_Analysis.ipynb

- 다른 토큰들 → [SEP]
  - ✓ Ex) Head 8-10 : direct objects(직접 목적어) → verbs
  - ✓ 대부분의 non-noun 단어들이 [SEP]로 향함
  - ✓ 이러한 관계는 'no-op', 혹은 '관계 없음' 이라고 분류

- Gradient-based 중요도 파악 (Sundararajan et al., 2017)
  - ✓ Attention weight에 대한 MLM loss의 미분 값의 평균 계산
  - ✓ Layer 6 -10 gradient가 매우 작음
  - ✓ [SEP]는 no-op 을 위한 토큰임을 보여줌



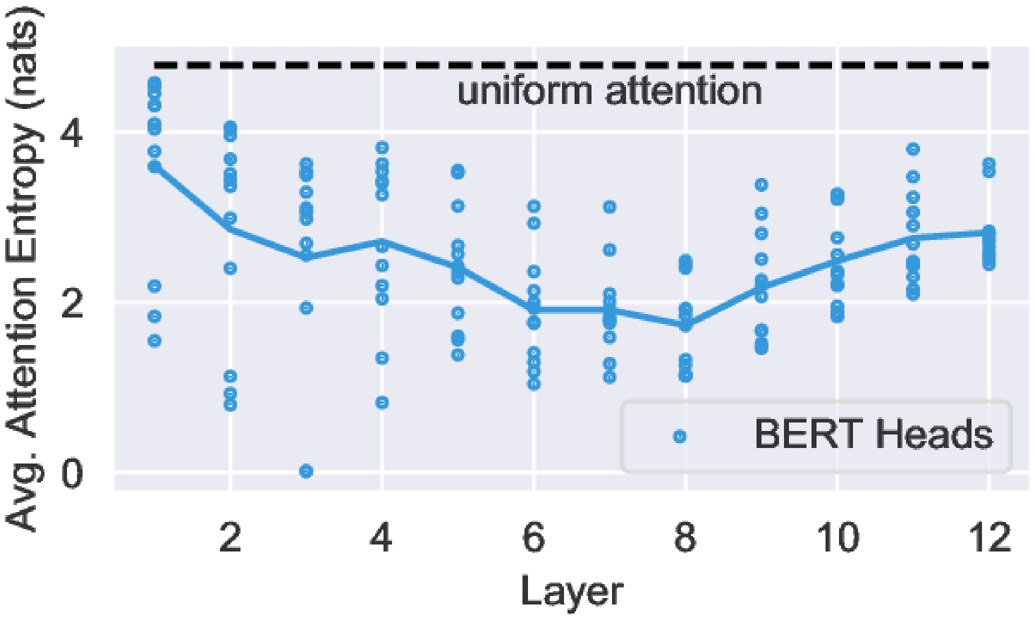



# 03 Surface-Level Patterns in Attention

Focused v.s Broad Attention

https://github.com/clarkkev/attention-analysis/blob/master/General_Analysis.ipynb

- Attention distribution의 Entropy를 통해 특정 단어에 attention이 몰렸는지 넓게 분포되어 있는지 확인
  - ✓ Entropy가 클수록 broad 함
  - ✓ Entropy가 작을수록 특정 단어로 focus 됨




- 초기 layer의 평균 attention entropy 가 높음
  - ✓ 한 단어로 향하는 attention 값이 전체의 최대 10% 밖에 안 됨
  - ✓ Attention 의 형태가 마치 bag-of-vectors 처럼 보임

- [CLS]으로 부터 나오는 attention entropy 확인
  - ✓ 특히 마지막 layer에서 entropy가 높은 값을 가짐
  - ✓ Broad하게 단어들을 살피며 aggregating 하고 있는 것임
  - ✓ [CLS]이 next-sentence-prediction 의 input으로 활용 됨 이 확인 됨

## 4 Probing Individual Attention Heads

# 04 Probing Individual Attention Heads

## Probing

- Probe
  - ✓ n. ⑤ 엄밀한 조사, 정사(精査); 탐사 / vi. 면밀히 조사하다; (미지의 세계·넓은 사막 등에) 들어가다

- What kind of linguistic information neural networks are able to capture?

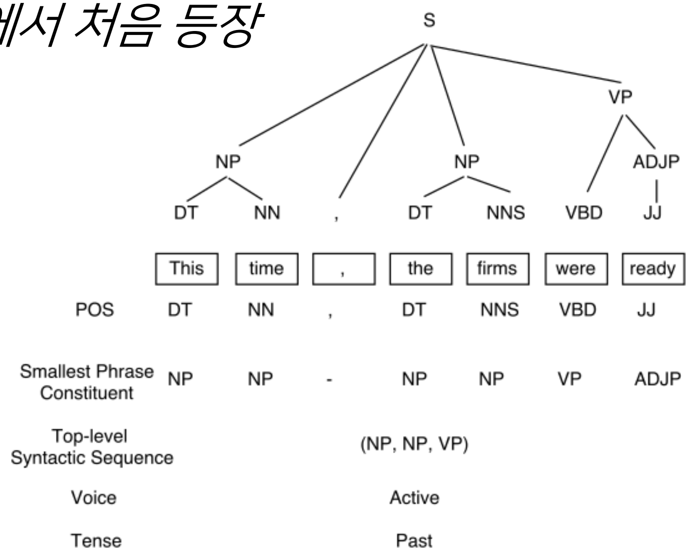
- Probing task
  - ✓ Use the encoded representations of one system to train another classifier on some other (probing) task of interest.
  - ✓ Probing classifier : A classifier on the encoded representations to predict external linguistic properties
  - ✓ To isolate some linguistic phenomena and if the probing classifier performs well on the probing task we infer that the system has encoded the linguistic phenomena in question
  - ✓ Linguistic phenomena : POS Tagging, Dependency Parsing, Coreference 등

https://spraakbanken.gu.se/blogg/index.php/2019/11/16/what-are-probing-tasks-in-nlp/#:~:text=Probing%20tasks%2C%20which%20have%20also,(probing)%20task%20of%20interest

# 04 Probing Individual Attention Heads

## Probing

- *Does String-Based Neural MT Learn Source Syntax?(Shi et al.,2016)에서 처음 등장*
  - ✓ Two-layer LSTM으로 이루어진 Seq2seq 모델 활용하여 NMT 진행
  - ✓ Encoder를 통해 sentence의 representation 으로 활용되는 cell states를 구함
  - ✓ 5 종류의 syntactic label을 지정함
  - ✓ Logistic Regression에 cell states를 입력하여 syntactic label을 예측




- *What you can cram into a single vector:*
*Probing sentence embeddings for linguistic properties*
*(Conneau et al., 2018)*
  - ✓ 10 probing tasks as a part of SentEval to evaluate
  the linguistic capabilities of sentence embeddings


- 본문에서는 Dependency parsing, coreference resolution의 probing task 진행

- Word-level tasks를 수행하기 위해 token-token attention map을 word-word로 바꿈
  - ✓ Attention 'To' split up word : Attention 값의 sum-up
  - ✓ Attention 'From' split up word : Attention 값의 평균

04

# Probing Individual Attention Heads

## Overview

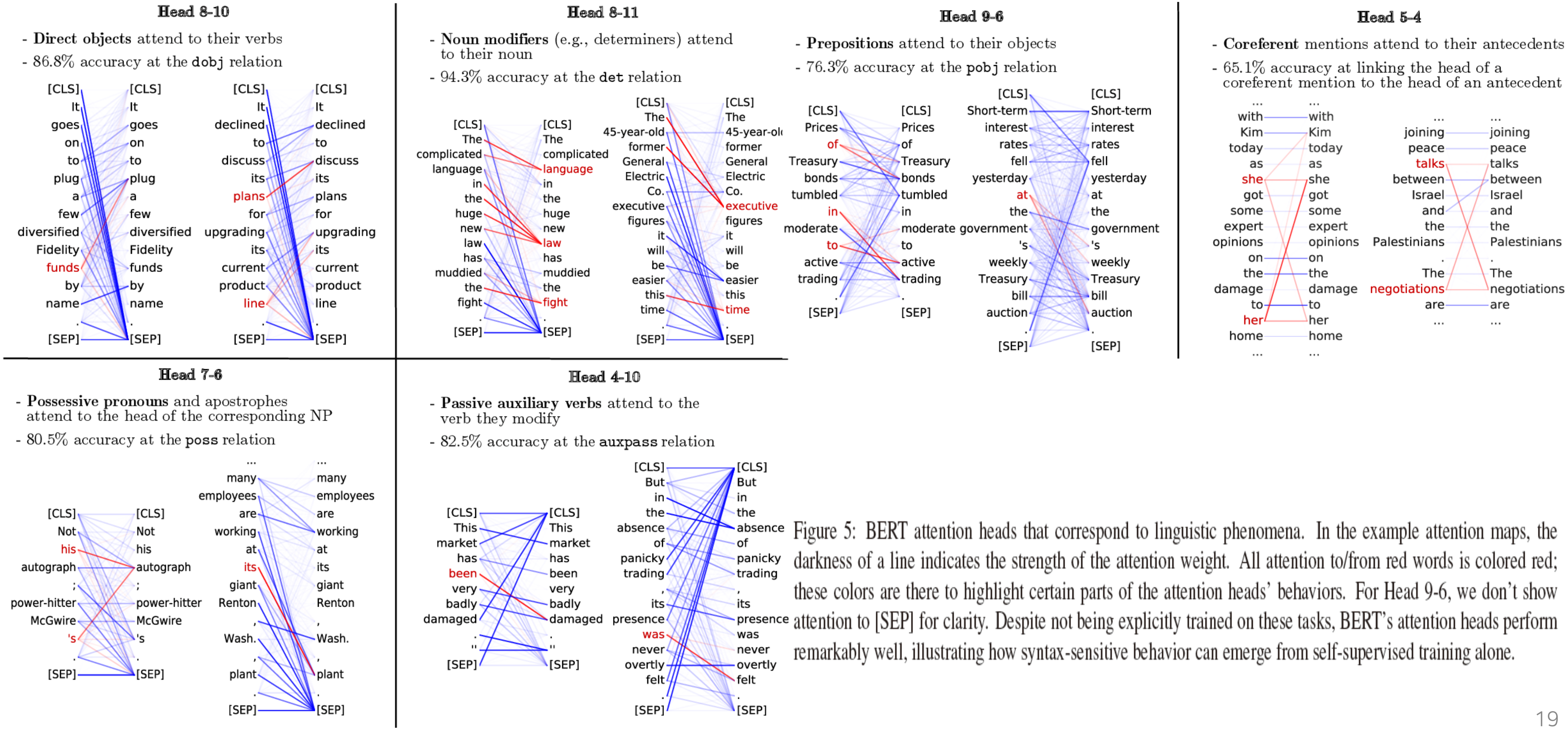
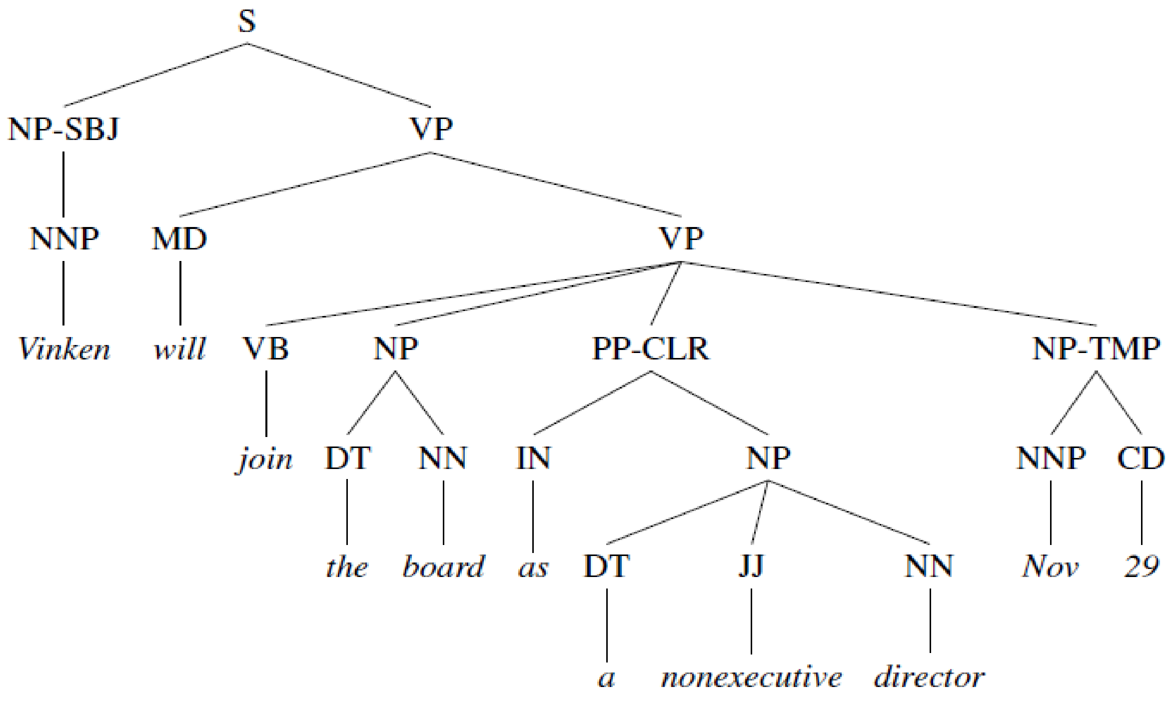



Figure 5: BERT attention heads that correspond to linguistic phenomena. In the example attention maps, the darkness of a line indicates the strength of the attention weight. All attention to/from red words is colored red; these colors are there to highlight certain parts of the attention heads' behaviors. For Head 9-6, we don't show attention to [SEP] for clarity. Despite not being explicitly trained on these tasks, BERT's attention heads perform remarkably well, illustrating how syntax-sensitive behavior can emerge from self-supervised training alone.

## 04 Probing Individual Attention Heads

### Dependency Syntax

- Setup
  - ✓ Wall Street Journal(WSJ) dependency parsing using Stanford Dependencies
  - ✓ Evaluating both directions : Head (dependency) →Dependent, Dependent→Head (dependency)
  - ✓ Simple fixed-offset baseline : ex) (-2) = dependent 단어 기준 왼쪽 두번째가 Head라고 여김




WSJ의 Dependency parsing 예시

| Relation | Examples with *head* and **dependent** |
|---|---|
| NSUBJ | **United** *canceled* the flight. |
| DOBJ | United *diverted* the **flight** to Reno. |
| | We *booked* her the first **flight** to Miami. |
| IOBJ | We *booked* **her** the flight to Miami. |
| NMOD | We took the **morning** *flight*. |
| AMOD | Book the **cheapest** *flight*. |
| NUMMOD | Before the storm JetBlue canceled **1000** *flights*. |
| APPOS | *United*, a **unit** of UAL, matched the fares. |
| DET | **The** *flight* was canceled. |
| | **Which** *flight* was delayed? |
| CONJ | We *flew* to Denver and **drove** to Steamboat. |
| CC | We flew to Denver **and** *drove* to Steamboat. |
| CASE | Book the flight **through** *Houston*. |

**Figure 14.3** Examples of core Universal Dependency relations.

https://web.stanford.edu/~jurafsky/slp3/14.pdf

## 04 Probing Individual Attention Heads

Dependency Syntax

- Setup
  - ✓ Wall Street Journal(WSJ) dependency parsing using Stanford Dependencies
  - ✓ Evaluating both directions : Head (dependency) →Dependent, Dependent→Head (dependency)
  - ✓ Simple fixed-offset baseline : Ex) (-2) = dependent 단어 기준 왼쪽 두번째가 Head라고 가정

- No single attention head does well at syntax 'overall'

- Except 'pobj', the dependent attends to the head word rather than the other way around
  - ✓ Dependent는 하나의 head를 갖지만, Head는 여러 dependent를 가짐

- 기존의 annotation이랑은 다른 예측을 할 수 있음
  - ✓ Ex) Attn Head 7-6, " 's " 를 poss 관계의 dependent 로 marking 함 원래는 " 's "앞의 단어가 dependent 로 되어 있음
  - ✓ By-product of self-supervised learning

- Attention head 별로 잘 파악하는 syntactic relation이 존재

| Relation | Head | Accuracy | Baseline |
|---|---|---|---|
| All | 7-6 | 34.5 | 26.3 (1) |
| prep | 7-4 | 66.7 | 61.8 (-1) |
| pobj | 9-6 | **76.3** | 34.6 (-2) |
| det | 8-11 | **94.3** | 51.7 (1) |
| nn | 4-10 | 70.4 | 70.2 (1) |
| nsubj | 8-2 | 58.5 | 45.5 (1) |
| amod | 4-10 | 75.6 | 68.3 (1) |
| dobj | 8-10 | **86.8** | 40.0 (-2) |
| advmod | 7-6 | 48.8 | 40.2 (1) |
| aux | 4-10 | 81.1 | 71.5 (1) |
| poss | 7-6 | **80.5** | 47.7 (1) |
| auxpass | 4-10 | **82.5** | 40.5 (1) |
| ccomp | 8-1 | **48.8** | 12.4 (-2) |
| mark | 8-2 | **50.7** | 14.5 (2) |
| prt | 6-7 | **99.1** | 91.4 (-1) |

# 04 Probing Individual Attention Heads

## Coreference Resolution

- Setup
  - ✓ CoNll-2012 Dataset 활용(Pradhan et al., 2012)
  - ✓ Compute antecedent selection accuracy
  - ✓ Baselines:
    1) Picking the nearest other mention
    2) Picking the nearest other mention with the same head word as the current mention
    3) Full string match, head word match, number/gender/person match, all other mentions 중 한 가지 조건이라도 만족시키는 nearest mention(*Lee et al.,2011)
  - ✓ Neural coreference system 도 같이 비교 (**Wiseman et al., 2015)

- Results
  - ✓ 꽤나 정확도가 높음
  - ✓ String match와 비교했을 때는  정확도가 10% 더 높음
  - ✓ Capable of fuzzy matching between synonyms

| Model | All | Pronoun | Proper | Nominal |
|---|---|---|---|---|
| Nearest | 27 | 29 | 29 | 19 |
| Head match | 52 | 47 | 67 | 40 |
| Rule-based | 69 | 70 | 77 | 60 |
| Neural coref | 83* | – | – | – |
| Head 5-4 | 65 | 64 | 73 | 58 |

*Only roughly comparable because on non-truncated documents and with different mention detection.

*Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task (CoNLL, 2011)
**Learning anaphoricity and antecedent ranking features for coreference resolution (ACL, 2015)

# 5 Probing Attention Head Combinations

# 05 Probing Attention Head Combinations

## Attention-based probing classifiers

- Model's overall knowledge about syntax is distributed across multiple attention heads

- BERT attention outputs as fixed
  - ✓ Not back-propagating into BERT, but only train a small number of parameters

- Classifier는  문장 내 어떤 단어가 주어진 단어의 syntactic head가 되는지 확률을 출력함

# 05 Probing Attention Head Combinations

## Attention-based probing classifiers

- Two types of classifier
  - ✓ Attention-Only Probe (Probe는 probing classifier, probing model 정도로 이해)

$$p(i|j) \propto \exp(\sum_{k=1}^{n} w_k a_{ij}^k + u_k a_{ji}^k)$$

    - ▫ Learning simple linear combination of attention weights
    - ▫ $p(i|j)$ : The probability of word $i$ being word $j$'s syntactic head
    - ▫ $\alpha_{ij}^k$ : The attention weight from word $i$ to $j$ produced by head $k$
    - ▫ $n$ : The number of attention heads
    - ▫ Include both directions : candidate head ↔ dependent
    - ▫ $w, u$ : weight vectors, trained using standard supervised learning

  - ✓ Attention-and-Words Probe

$$p(i|j) \propto \exp(\sum_{k=1}^{n} W_{k,:}(v_i \oplus v_j) a_{ij}^k + U_{k,:}(v_i \oplus v_j) a_{ji}^k)$$

    - ▫ $v$ : GloVe embedding vector / $\oplus$ : Concatenation
    - ▫ $W, U$ : Weight matrices
    - ▫ $W_{k,:}(v_i \oplus v_j)$ : Word-sensitive weight for the particular attention head

# 05 Probing Attention Head Combinations

## Three Baselines and Results

1. Right-branching
   - ✓ Always predicts the head is to the dependent's right
2. Simple one-hidden-layer network
   - ✓ Input: Glove embeddings for the dependent and candidate head, distance features between  the two words
3. Attention-and-words probe
   - ✓ Attention maps from a BERT network with pretrained word/positional embedding but randomly initialized other weights

| Model | UAS |
|---|---|
| Structural probe | 80 UUAS* |
| Right-branching | 26 |
| Distances + GloVe | 58 |
| Random Init Attn + GloVe | 30 |
| Attn | 61 |
| Attn + GloVe | 77 |

- BERT learns some aspects syntax purely as a by-product of self-supervised training
- Rich한 pre-training 으로도 언어의 계층적 구조를 충분히 파악할 수 있음

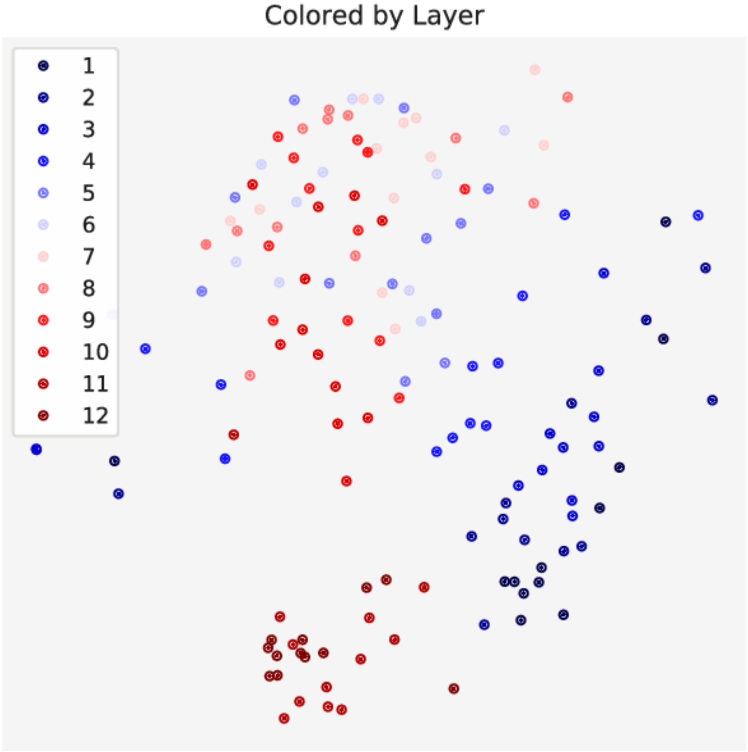
# 6 Clustering Attention Heads

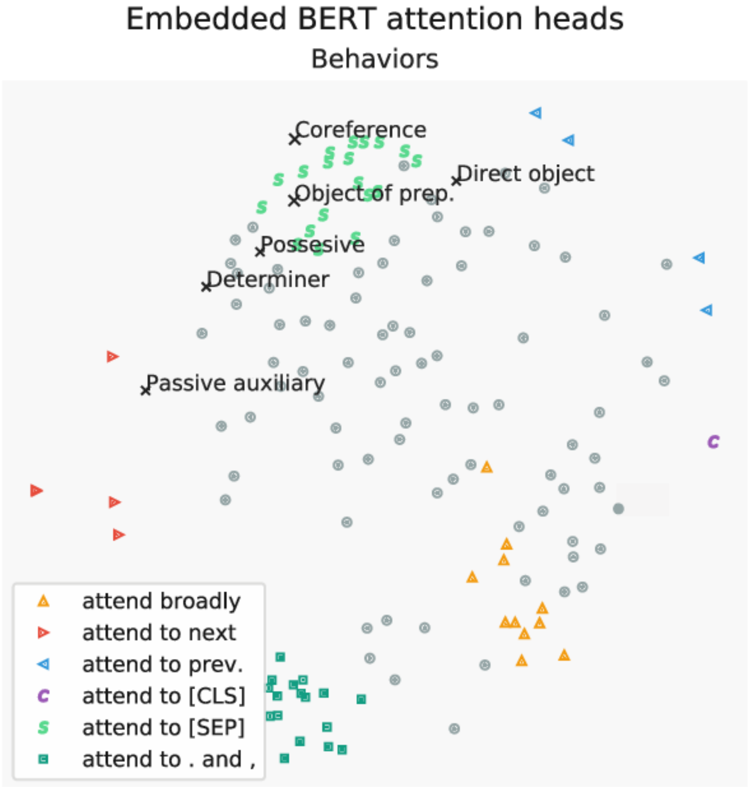
# 06 Clustering Attention Heads

## Three Baselines and Results

- Jensen-Shannon Divergence를 기준으로 클러스터링 진행

$$\sum_{token \in data} JS(H_i(token), H_j(token))$$



## 7 Related Work

# 07 Related Work

## Understanding what neural networks learn

- Examining the outputs of language models on carefully chosen input sentences
  - ✓ Ex) 주어의 동사 찾기

- Investigating the internal vector representations of the model using probing classifiers
  - ✓ Probing classifier
    : vector representation을 입력 받아 지도학습 task(Ex) POS Tagging 등)를 수행하는 간단한 인공신경망
  - ✓ Probing classifier의 성능이 좋다는 뜻은 vector representation 이 언어정보를 잘 반영한다는 뜻

- Analyzing attention
  - ✓ Attention <u>is not</u> explanation (Jain and Wallace, 2019)
    - ▪ Attention weights do not correlate with other measures of feature importance
    - ▪ Attention weights can be changed without altering model predictions
  - ✓ Attention <u>is not not</u> explanation (Wiegreffe and Pinter, 2019)
    - ▪ attention is explanation depends on the definition of explainability one is looking for

[2021 DSBA 세미나] 윤훈상 석사과정 발표 영상 (https://youtu.be/8VeWItX9GhI)

# 감사합니다