# Neural Networks and Deep Learning
## Transformer & Bert

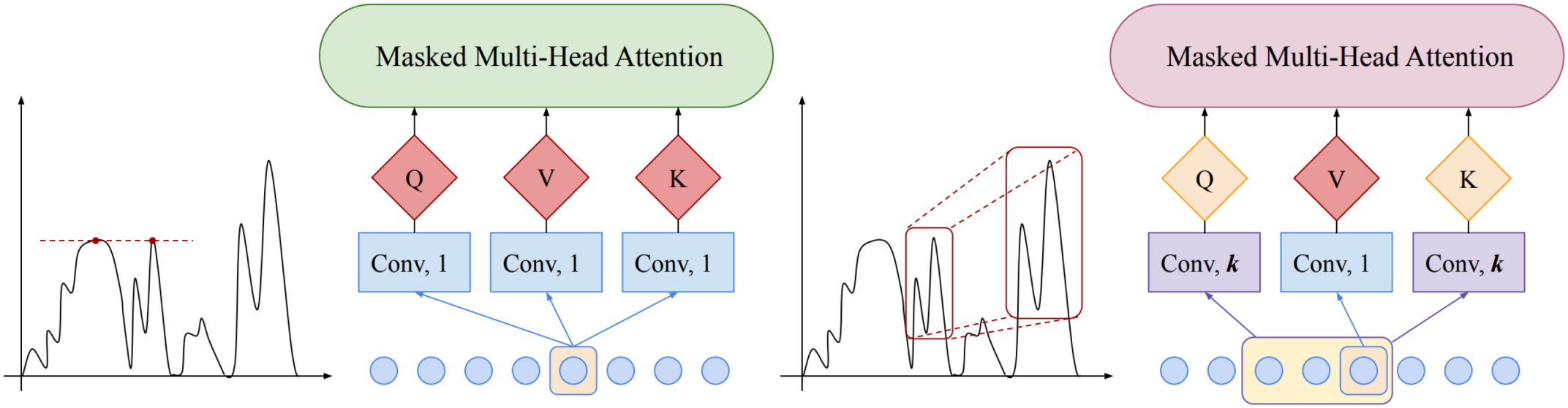
Shumin Wu
Department of Computer Science
shumin.wu@colorado.edu
March 4, 2020

# Previous Lecture: Transformer Variation

- Transformer implementations
  - tensor2tensor
  - Trax
  - Tensorflow transformer tutorial (Keras)
- Recurrence (Universal Transformer)
  - Shared weights between encoder blocks and between decoder blocks
  - Perform more iterations on more ambiguous tokens (dynamic halting)
- Long-sequence attention
  - Key & value convolution w/ stride
  - Local (fixed window size) attention
  - Query/key convolution (to detect local patterns for time series)
  - LogSparse attention: does not directly attend to all previous tokens

# Time Series: Attention Locality

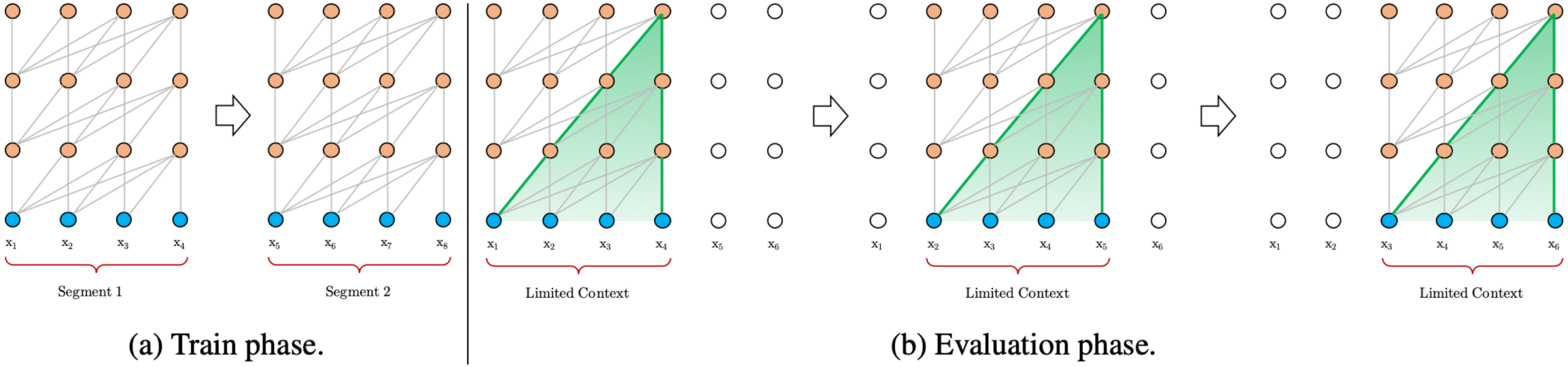
Convolution of Q, K for attention distribution that has stronger locality awareness




source: Li et al., 2019

# Local Attention

Normal segmented attention:




source: Dai et al., 2019

A lot of similar computation at decoding time, without benefit of long-range attention.

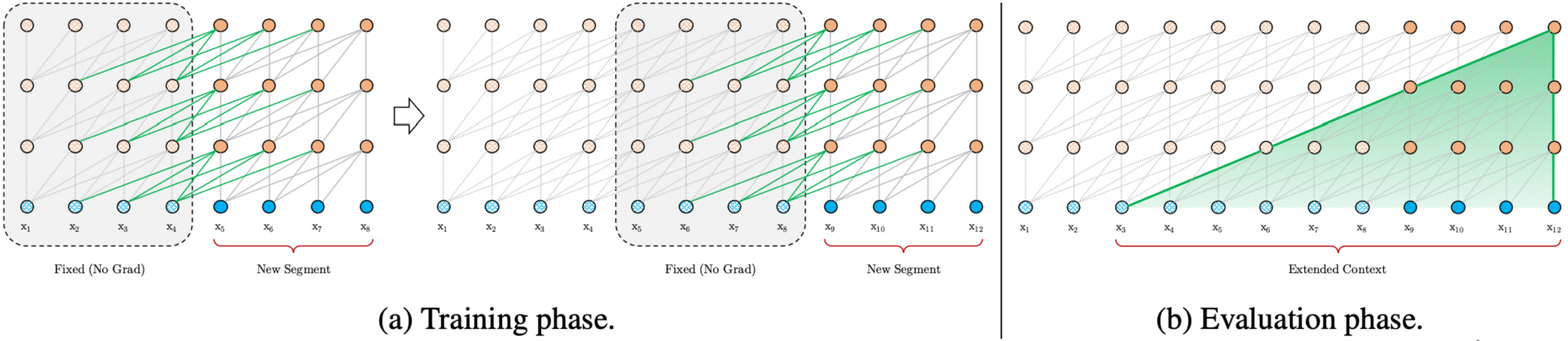
# Language Model: Transformer XL

Attention block of current segment computed with previous attention output of current segment and previous segment, but only backprop on current segment:

$$\widetilde{\mathbf{h}}_{\tau+1}^{n-1} = [\text{stop-grad}(\mathbf{h}_{\tau}^{n-1}) \;\; \mathbf{h}_{\tau+1}^{n-1}]$$

$$\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} = \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_{q}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{k}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{v}^{\top}$$

Achieves longer dependency at each higher attention block.




source: Dai et al., 2019

# Scaled Dot-Product Attention

Recall transformer attention function:

$$C = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

*softmax* output is dominated by the largest elements.

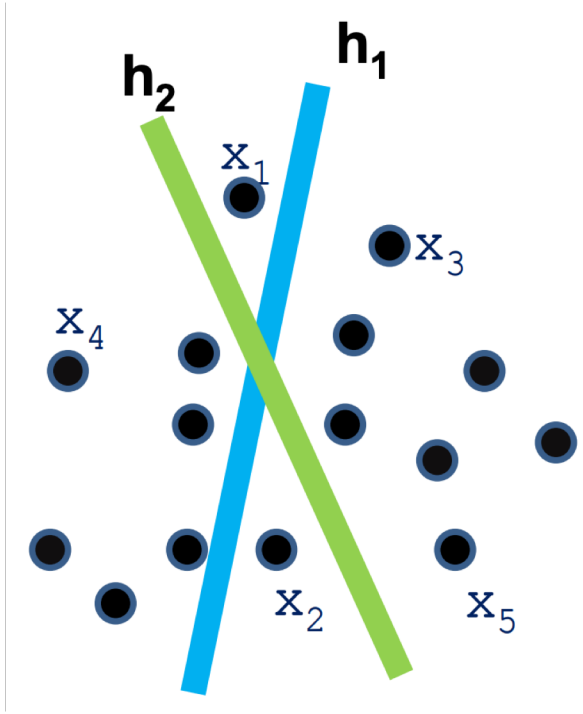
Attention output can be approximated with the top matching key vectors for each query vector.

For a given vector, if we can just efficiently find the *top m* vectors (say linearly), then we can compute attention in $O(m \cdot n \cdot d)$.

Where have we seen this problem before?

# Approximate Nearest Neighbor: Hash

Linear projection based partitioning




| X | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $y_1$ | 0 | 1 | 1 | 0 | 1 |
| $y_2$ | 1 | 0 | 1 | 0 | 1 |
| … | … | … | … | … | … |
| $y_m$ | … | … | … | … | … |
| | 010… | 100… | 111… | 001… | 110… |

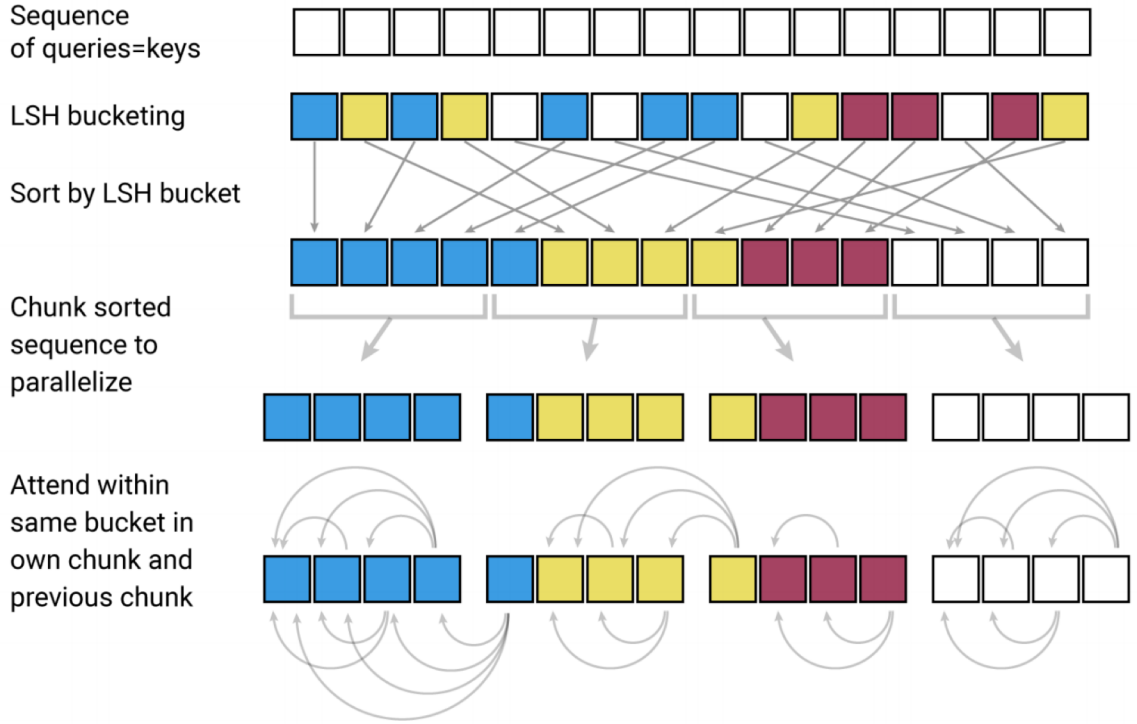
Source: [Large-Scale Machine Learning, EECS 6898]

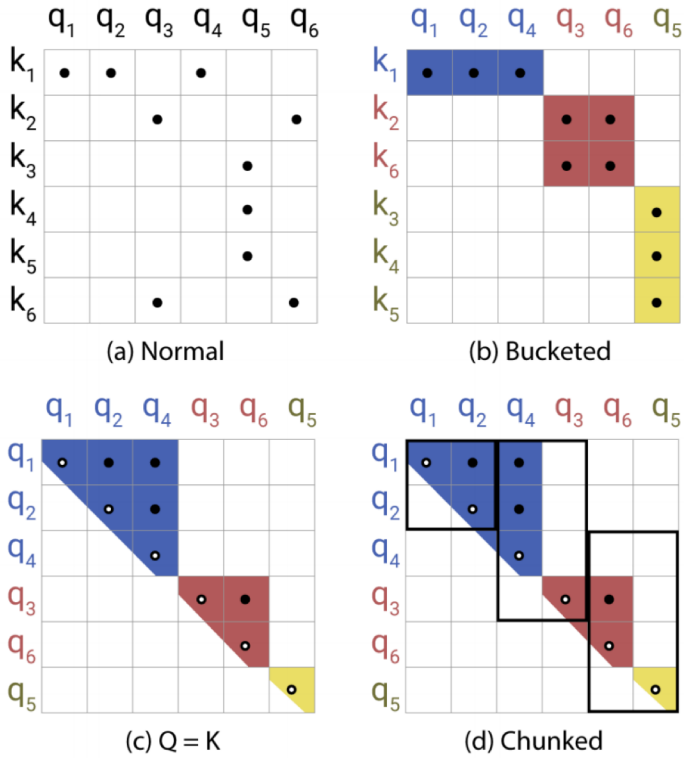
# Reformer: Query/Key Attention Hashing

Basic idea: use LSA to efficiently compute attention distribution

More optimization:

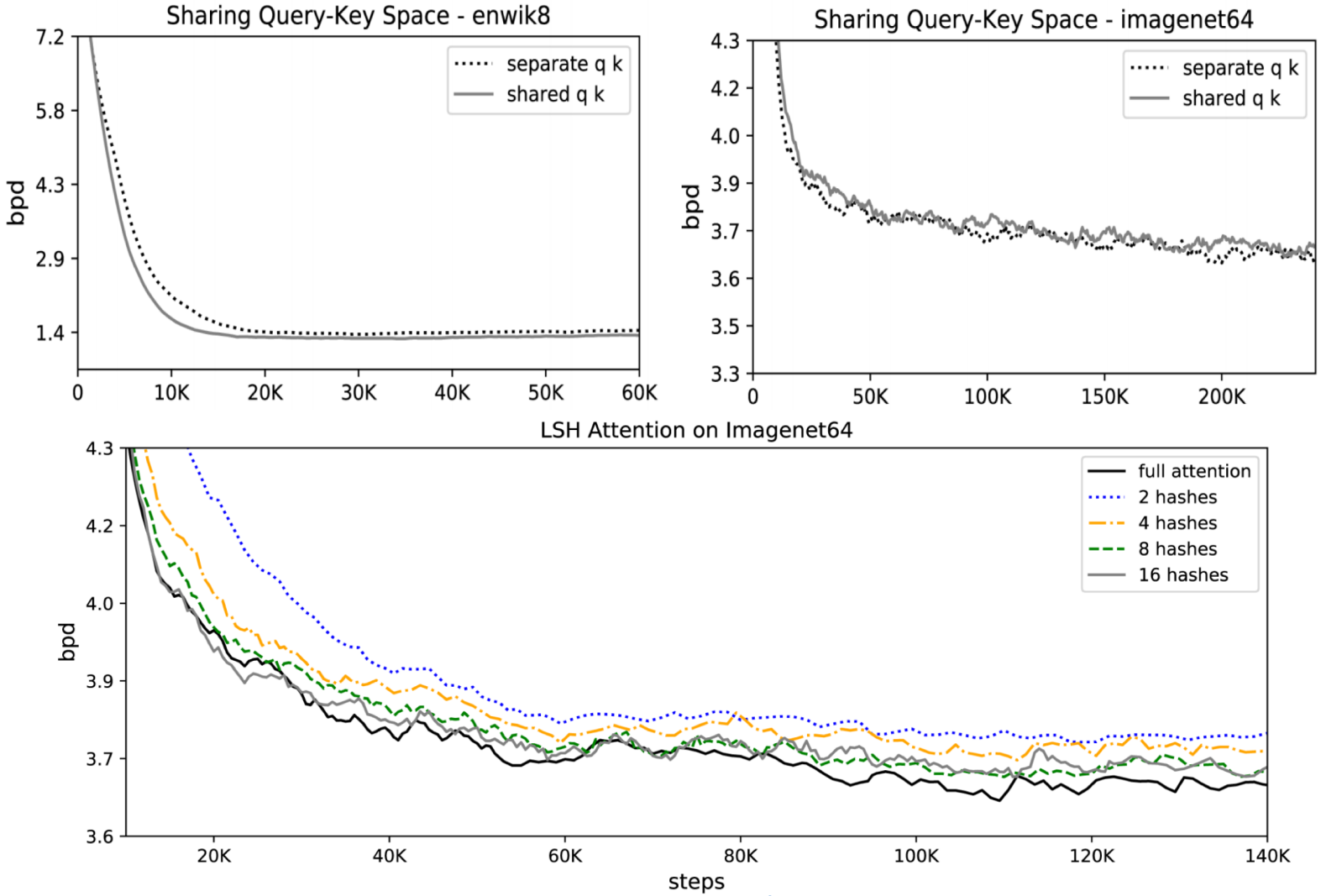
- Query == Key
- Per-batch index
- Multiple hashing rounds
- attention within 2 adjacent chunks




source: [Kitaev et al., 2020]

# Reformer: Ablation




source: [Kitaev et al., 2020]

# Reformer: Image Completion

source: [Google AI blog]

# Previously: Language Model

- Modeling the probability of a sequence of text in a given language
- Applications
  - Speech recognition
  - Intelligent keyboard
  - Smart reply
  - Machine translation

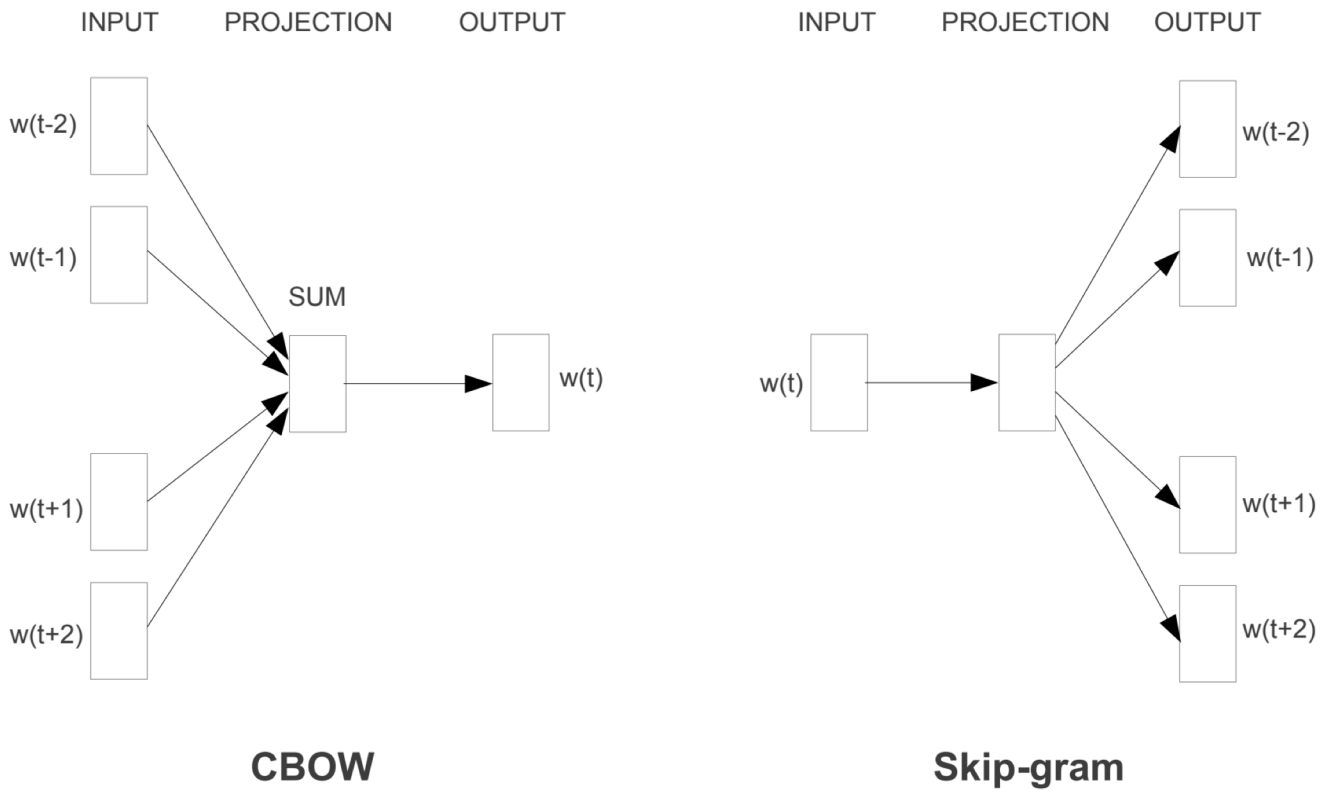
Why do cats

*hate water*
*read books*
*neural network*

*Training data for a language model is typically widely available: just need unlabeled text*

# Previously: Word2vec

- Learning word representation without explicit language modeling
  - Continuous bags of words (CBOW): given a window of surrounding words (before and after) without order, predict the missing middle word
  - Skip-gram: given the middle word, predict a window of surrounding word.
- Widely used as input to many NLP tasks.




source: [Mikolov et al., 2013]

# Contextual Representation

Word2Vec: learns a single representation for each word

Issue: words can take on different meaning in context:

open a bank account vs on the river bank

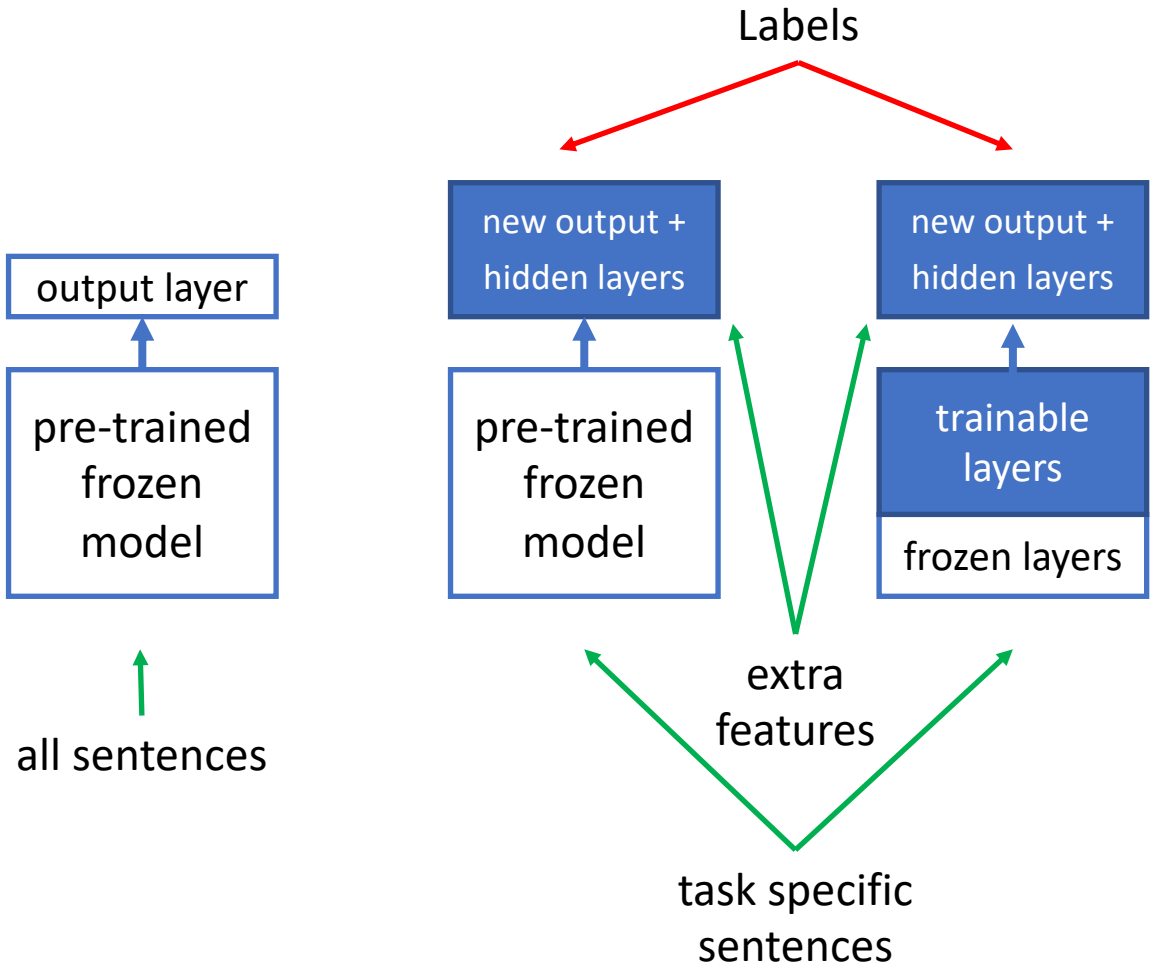
Solution: learn contextualized word representations

- RNN + LSTM (Semi-supervised Sequence Learning)
- RNN + bi-directional LSTM (ELMo):
  - Backward LSTM learns a “backward” language model, performs joint loss optimization with forward LSTM
  - Weight sharing of softmax activation & input embedding
- Transformer decoder (Improving Language Understanding by Generative Pre-Training)

Note: unlike word2vec, this requires having the pre-trained model to either fine-tune a task or extract a representation of the word sequence as input to the task (featured based)

# Previously: Pre-training and Fine-Tuning

- Start w/ pre-trained model
- Add new output layer to last layer of network
  - Add other layers in between if needed
- Train new layers w/ task specific data
- Unfreeze top (or all) layers of pre-trained model
  - Retrain all trainable layers w/ task specific data



# Unidirectional Language Model

Example:

i think teddy _____ is a great president
i think teddy _____ make great presents for kids

In either the forward or backward direction, the missing word is ambiguous.

But in the context of all other words in the sentence, the answer is obvious.

Recall: word2vec is trained on a surrounding window (bag) of words.

# BERT: **B**idirectional **E**ncoder **R**epresentations from Transformers

- LM based on self-attention of whole sentence

  Issue: what is there to learn if the whole sentence is available?

  Solution: mask some of the words in the input

  **Input**: i think teddy [MASK] make great [MASK] for [MASK]

  **Prediction**: i think teddy bears make great presents for kids

- Masked LM needs to balance training time and context availability
  - Too little masking: takes a long time to train
  - Too much masking: not enough surrounding context to predict missing words

# BERT Masked LM

Mask 15% of random input words

Later work found masking consecutive tokens (phrase) is helpful

Issue: [MASK] token is not used in inputs of actual tasks

Solution: replace with random words some of the times

- 80%: i think teddy bears → i think teddy [MASK]
- 10%: i think teddy bears → i think teddy bears
- 10%: i think teddy bears → i think teddy buckle (random)

# BERT: Next Sentence Prediction

Train relationship between sentences:

**Sentence A**: The car broke down on the side of the road.
**Sentence B**: It had to be towed to the nearest repair shop.
**Label**: IsNextSentence

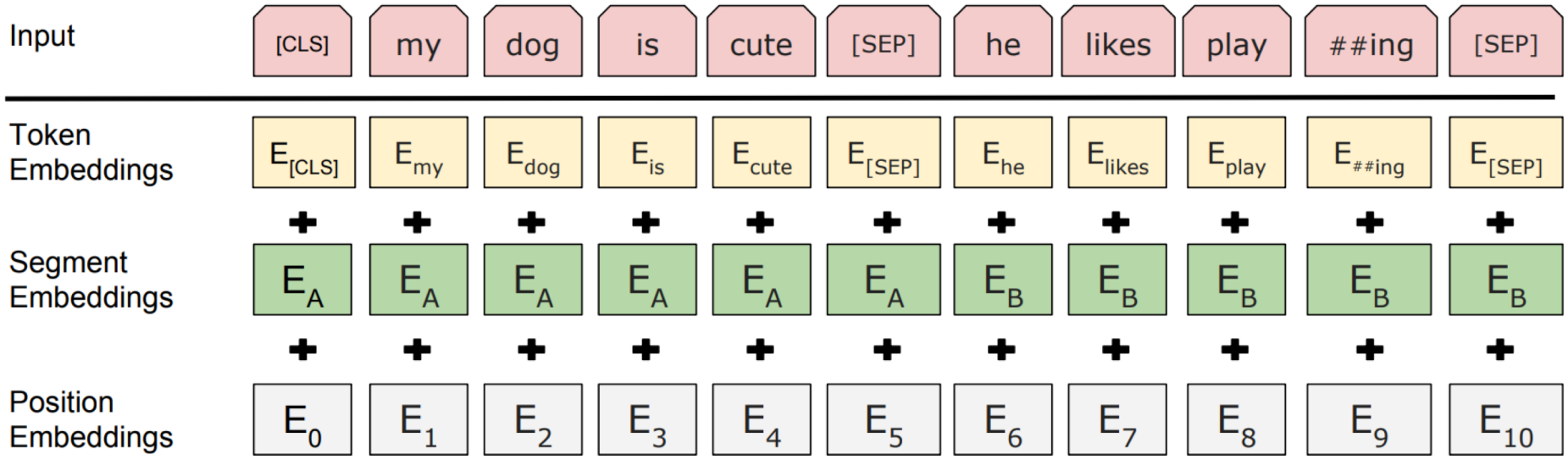
**Sentence A**: Ernie is an orange muppet character.
**Sentence B**: Google is leveraging BERT to better understand user searches.
**Label**: NotNextSentence

# BERT Input Representation

Trained sub-word encoding (30K vocabulary)

Transformer → T, ran, s, former

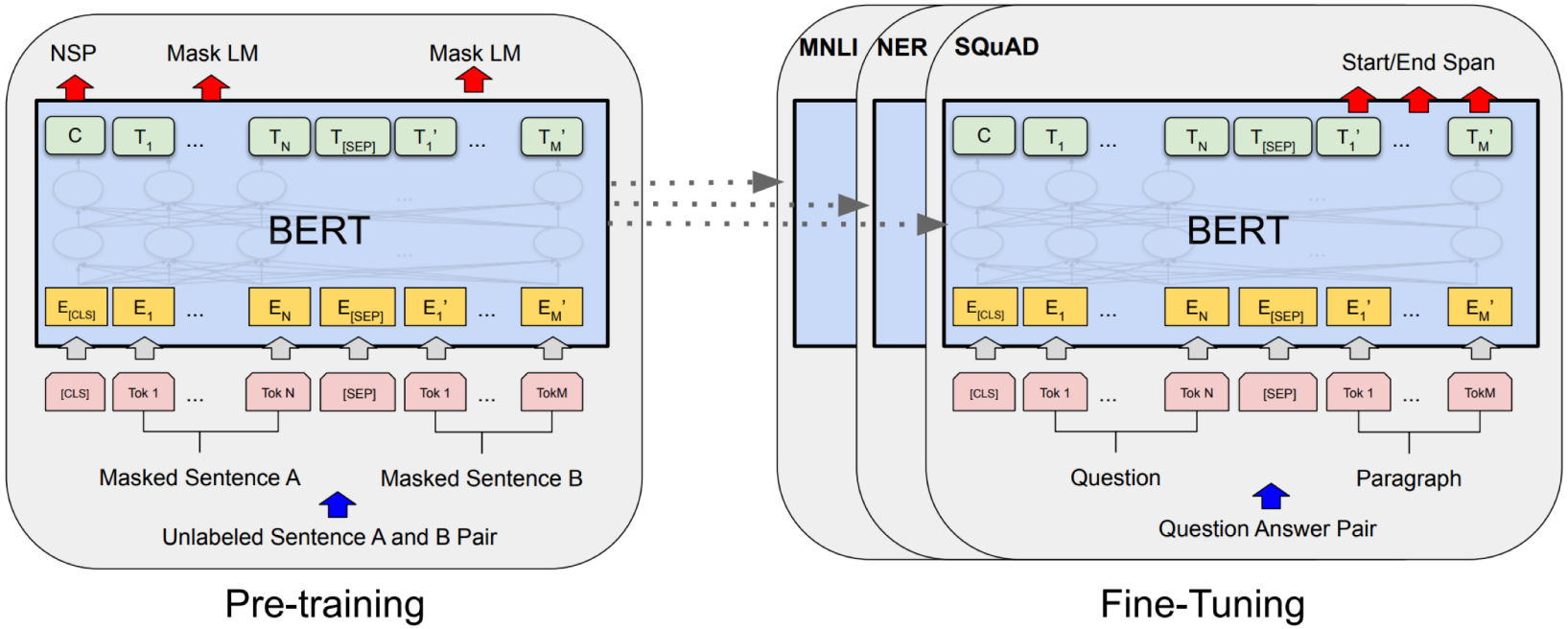
Embed sentence boundary w/ position




source: [Devlin et al., 2019]

# BERT: Pre-training and Fine-Tuning

- Using pre-trained Masked LM to fine-tune on NLU tasks
  - Achieved stat-of-the-art results on 11 GLUE tasks
  - Good performance on new NLP task w/o a lot of task-specific training data




source: [Devlin et al., 2019]

# BERT GLUE Performance

| System | MNLI-(m/mm) 392k | QQP 363k | QNLI 108k | SST-2 67k | CoLA 8.5k | STS-B 5.7k | MRPC 3.5k | RTE 2.5k | **Average** - |
|---|---|---|---|---|---|---|---|---|---|
| Pre-OpenAI SOTA | 80.6/80.1 | 66.1 | 82.3 | 93.2 | 35.0 | 81.0 | 86.0 | 61.7 | 74.0 |
| BiLSTM+ELMo+Attn | 76.4/76.1 | 64.8 | 79.8 | 90.4 | 36.0 | 73.3 | 84.9 | 56.8 | 71.0 |
| OpenAI GPT | 82.1/81.4 | 70.3 | 87.4 | 91.3 | 45.4 | 80.0 | 82.3 | 56.0 | 75.1 |
| $\text{BERT}_{\text{BASE}}$ | 84.6/83.4 | 71.2 | 90.5 | 93.5 | 52.1 | 85.8 | 88.9 | 66.4 | 79.6 |
| $\text{BERT}_{\text{LARGE}}$ | **86.7/85.9** | **72.1** | **92.7** | **94.9** | **60.5** | **86.5** | **89.3** | **70.1** | **82.1** |

source: Devlin et al., 2019

# BERT on SQuAD (Question/Answering)

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
**gravity**

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
**graupel**

Where do water droplets collide with ice crystals to form precipitation?
**within a cloud**

source: Rajpurkar et al., 2016

| System | Dev | | Test | |
|---|---|---|---|---|
| | EM | F1 | EM | F1 |
| Top Leaderboard Systems (Dec 10th, 2018) | | | | |
| Human | - | - | 82.3 | 91.2 |
| #1 Ensemble - nlnet | - | - | 86.0 | 91.7 |
| #2 Ensemble - QANet | - | - | 84.5 | 90.5 |
| Published | | | | |
| BiDAF+ELMo (Single) | - | 85.6 | - | 85.8 |
| R.M. Reader (Ensemble) | 81.2 | 87.9 | 82.3 | 88.5 |
| Ours | | | | |
| $\text{BERT}_{\text{BASE}}$ (Single) | 80.8 | 88.5 | - | - |
| $\text{BERT}_{\text{LARGE}}$ (Single) | 84.1 | 90.9 | - | - |
| $\text{BERT}_{\text{LARGE}}$ (Ensemble) | 85.8 | 91.8 | - | - |
| $\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA) | **84.2** | **91.1** | **85.1** | **91.8** |
| $\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA) | **86.2** | **92.2** | **87.4** | **93.2** |

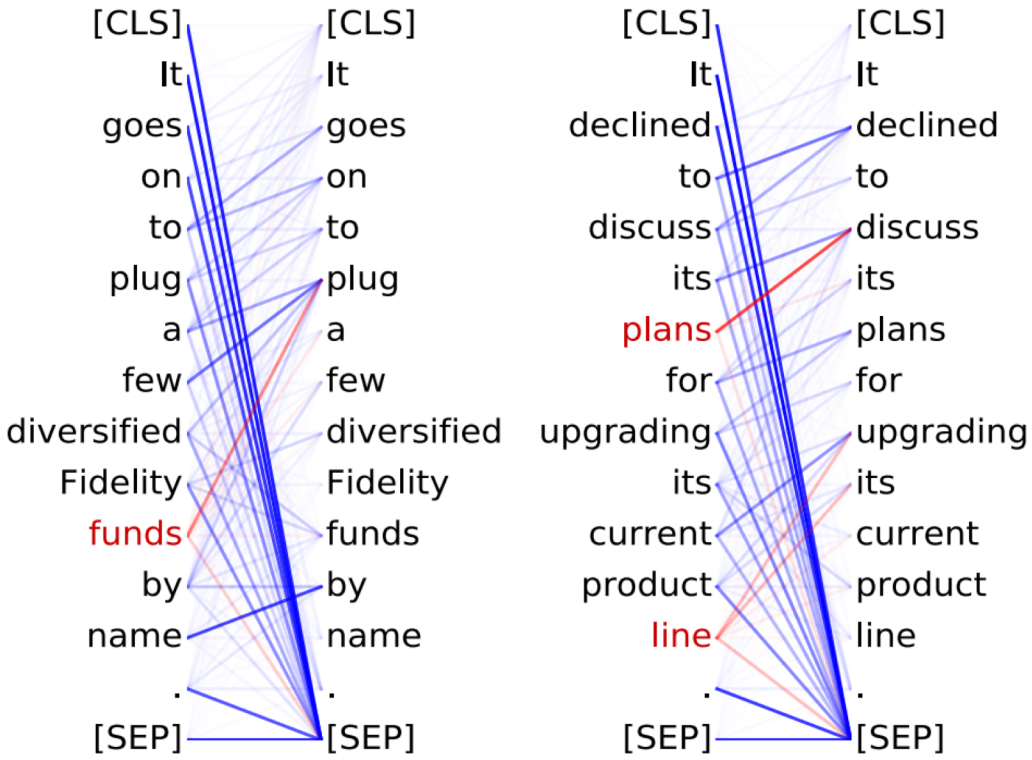
source: Devlin et al., 2019

Is BERT just really good at memorizing/guessing?

# BERT Attention Analysis

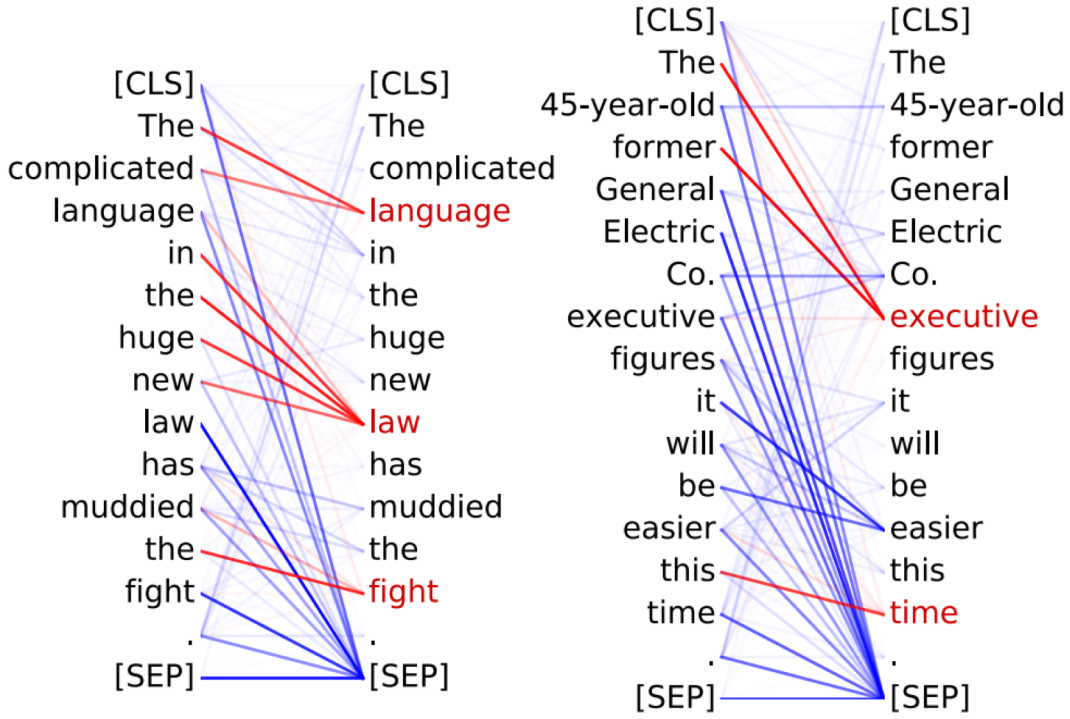
**Head 8-10**

- **Direct objects** attend to their verbs
- 86.8% accuracy at the `dobj` relation




**Head 8-11**

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the `det` relation

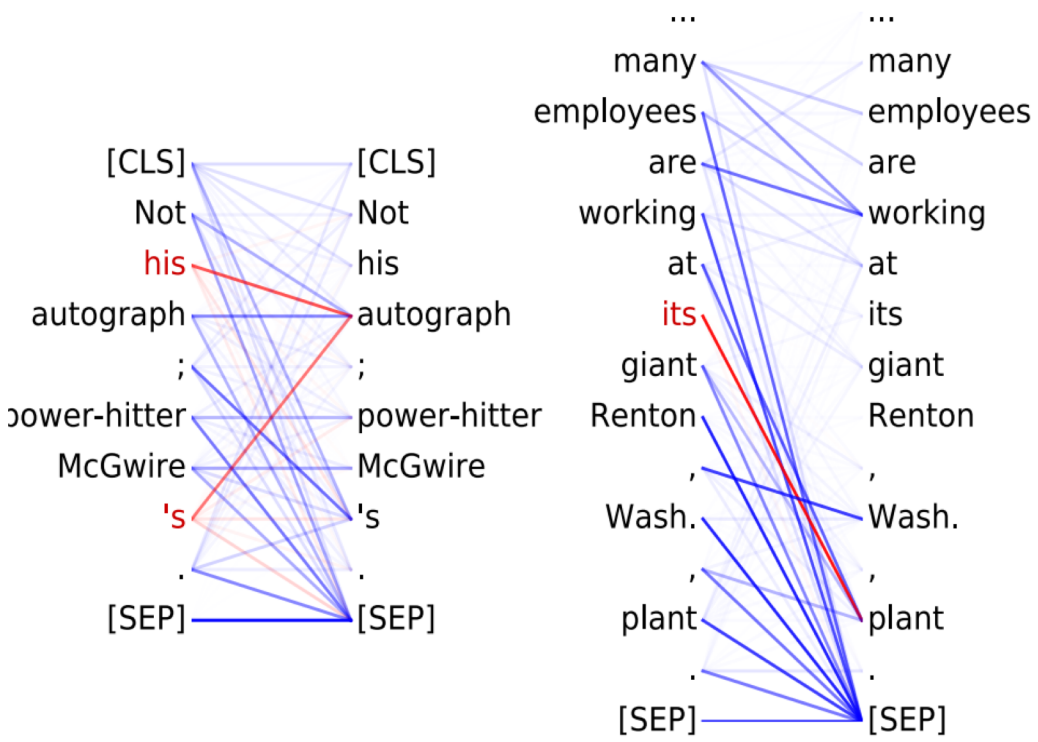



source: [Clark et al., 2019]

# BERT Attention Analysis

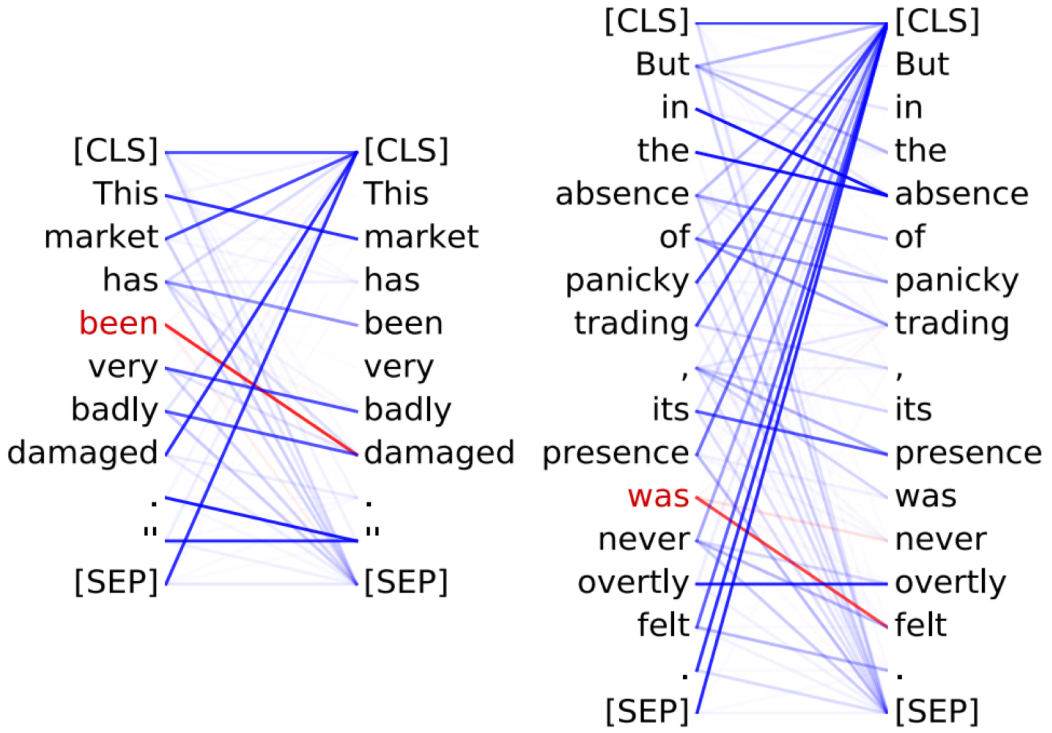
**Head 7-6**

- **Possessive pronouns** and apostrophes attend to the head of the corresponding NP
- 80.5% accuracy at the **poss** relation




**Head 4-10**

- **Passive auxiliary verbs** attend to the verb they modify
- 82.5% accuracy at the **auxpass** relation

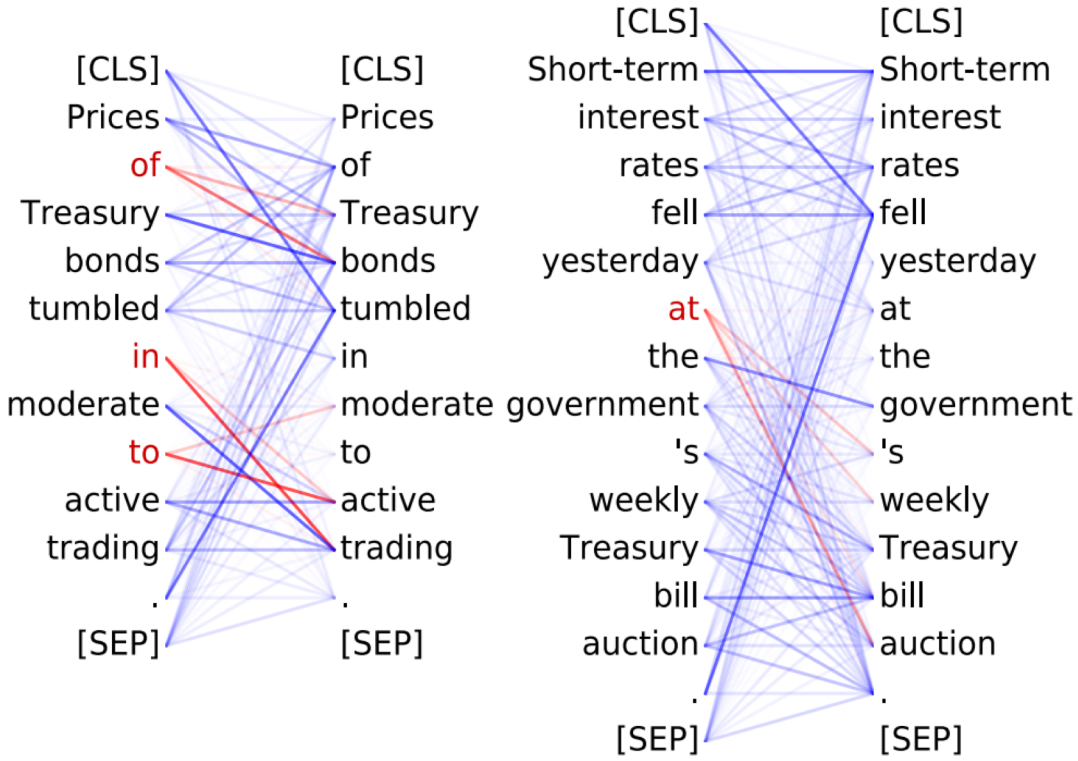



source: [Clark et al., 2019]

# BERT Attention Analysis

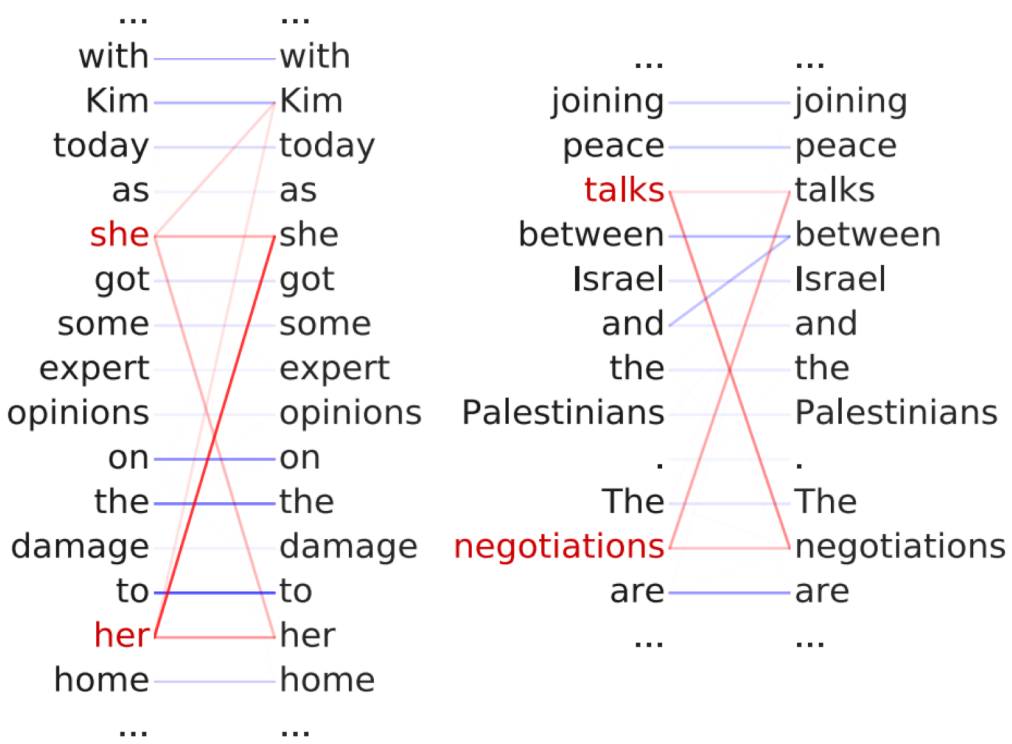
**Head 9-6**

- **Prepositions** attend to their objects
- 76.3% accuracy at the `pobj` relation




**Head 5-4**

- **Coreferent** mentions attend to their antecedents
- 65.1% accuracy at linking the head of a coreferent mention to the head of an antecedent




source: [Clark et al., 2019]

# BERT Follow on

- Transformer w/ shared weights/recurrence (ALBERT)
  - Smaller model
- Sentence ordering pretraining task: predict sentence ordering in the same document
  - BERT next sentence prediction found to be too easy
    - negative sentence pairs drawn from different documents
  - Binary decision: ALBERT, T5
  - Three way ($a$ follows $b$, $b$ follows $a$, neutral): StructBERT
- Many pretraining tasks (ERNIE)
  - Some are output of supervised models (discourse relation) or search engine output (IR relevance)

# GLUE Leaderboard (2020/03/05)

| Rank | Name | Model | URL | Score | CoLA | SST-2 | MRPC | STS-B | QQP | MNLI-m | MNLI-mm | QNLI | RTE | WNLI | AX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Alibaba DAMO NLP | StructBERT | | 90.3 | 75.3 | 97.1 | 93.9/91.9 | 93.0/92.5 | 74.8/91.0 | 90.9 | 90.7 | 96.4 | 90.2 | 94.5 | 49.1 |
| 2 | T5 Team - Google | T5 | | 90.3 | 71.6 | 97.5 | 92.8/90.4 | 93.1/92.8 | 75.1/90.6 | 92.2 | 91.9 | 96.9 | 92.8 | 94.5 | 53.1 |
| 3 | ERNIE Team - Baidu | ERNIE | | 90.1 | 72.8 | 97.5 | 93.2/91.0 | 92.9/92.5 | 75.2/90.8 | 91.2 | 90.8 | 96.1 | 90.9 | 94.5 | 49.4 |
| 4 | Microsoft D365 AI & MSR AI & GATECH | MT-DNN-SMART | | 89.9 | 69.5 | 97.5 | 93.7/91.6 | 92.9/92.5 | 73.9/90.2 | 91.0 | 90.8 | 99.2 | 89.7 | 94.5 | 50.2 |
| 5 | Microsoft D365 AI & UMD | FreeLB-RoBERTa (ensemble) | | 88.4 | 68.0 | 96.8 | 93.1/90.8 | 92.3/92.1 | 74.8/90.3 | 91.1 | 90.7 | 95.6 | 88.7 | 89.0 | 50.1 |
| 6 | Junjie Yang | HIRE-RoBERTa | | 88.3 | 68.6 | 97.1 | 93.0/90.7 | 92.4/92.0 | 74.3/90.2 | 90.7 | 90.4 | 95.5 | 87.9 | 89.0 | 49.3 |
| 7 | Facebook AI | RoBERTa | | 88.1 | 67.8 | 96.7 | 92.3/89.8 | 92.2/91.9 | 74.3/90.2 | 90.8 | 90.2 | 95.4 | 88.2 | 89.0 | 48.7 |
| 8 | Microsoft D365 AI & MSR AI | MT-DNN-ensemble | | 87.6 | 68.4 | 96.5 | 92.7/90.3 | 91.1/90.7 | 73.7/89.9 | 87.9 | 87.4 | 96.0 | 86.3 | 89.0 | 42.8 |
| 9 | GLUE Human Baselines | GLUE Human Baselines | | 87.1 | 66.4 | 97.8 | 86.3/80.8 | 92.7/92.6 | 59.5/80.4 | 92.0 | 92.8 | 91.2 | 93.6 | 95.9 | - |
| 10 | Stanford Hazy Research | Snorkel MeTaL | | 83.2 | 63.8 | 96.2 | 91.5/88.5 | 90.1/89.7 | 73.1/89.9 | 87.6 | 87.2 | 93.9 | 80.9 | 65.1 | 39.9 |

source: https://gluebenchmark.com/leaderboard

# What's Next?

- Pre-trained corpus biases toward our current understanding of the world
  - New discovery
  - Understanding in a fantasy world
- Practical application
  - Inference on mobile devices
  - Low resource languages
  - Translation without a lot of parallel text