A Brief Intro To BERT

Qiang Ning

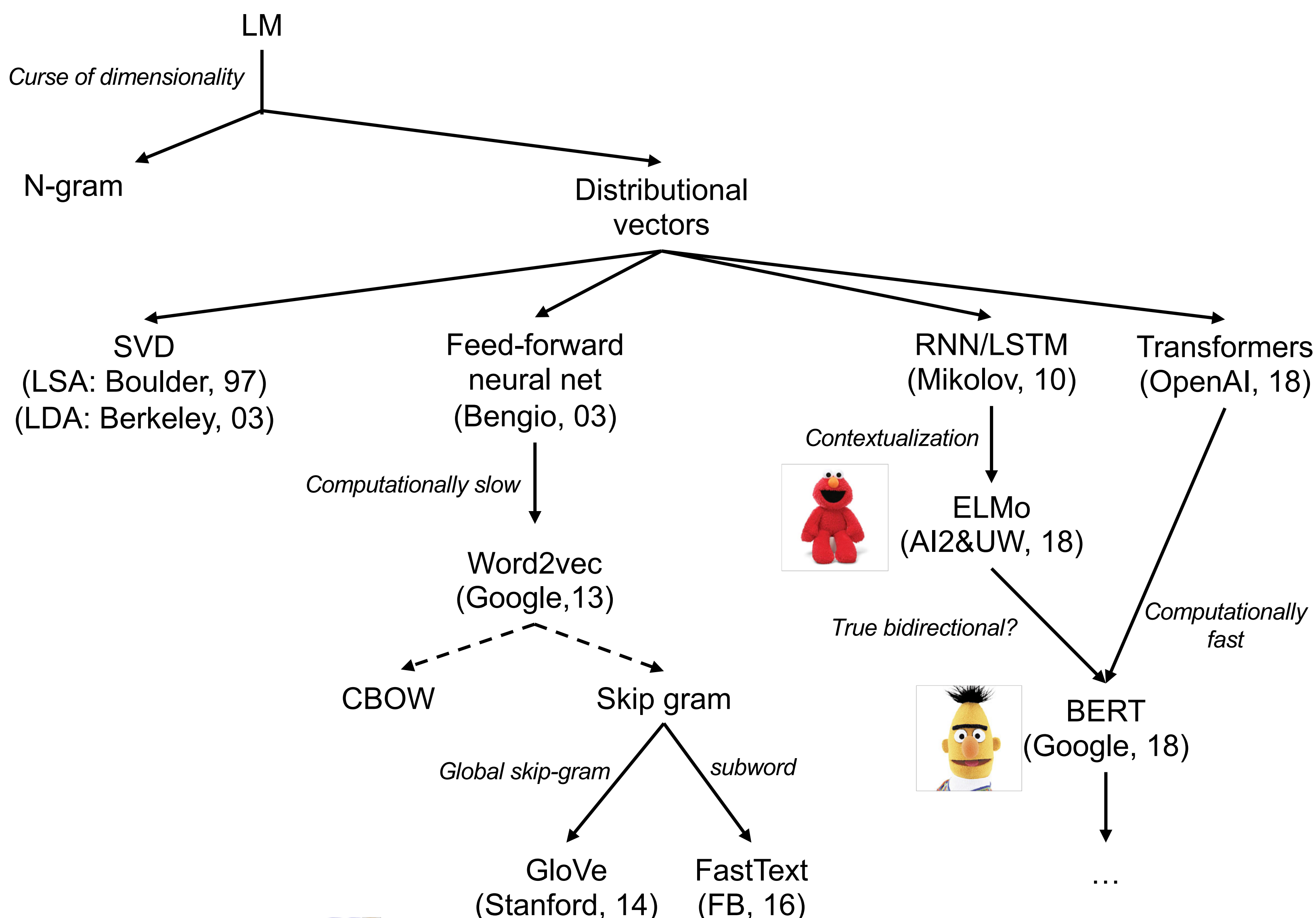
Presented at the C3SR weekly meeting

03/05/2019

Cognitive Computation Group

Roadmap Of Language Modeling

$$P(w_1, w_2, w_3, \ldots, w_t) = ?$$



Roadmap Of Language Modeling

$P(w_1, w_2, w_3, \dots, w_t) = ?$

LM

Curse of dimensionality

$\boxed{\text{N-gram}}$

$P(w_1, w_2, w_3) = P(w_1)P(w_2|w_1)P(w_3|w_2)$

Distributional vectors

SVD
(LSA: Boulder, 97)
(LDA: Berkeley, 03)

Feed-forward neural net
(Bengio, 03)

Computationally slow

Word2vec
(Google,13)

CBOW

Skip gram

Global skip-gram

subword

GloVe
(Stanford, 14)

FastText
(FB, 16)

RNN/LSTM
(Mikolov, 10)

Contextualization

ELMo
(AI2&UW, 18)

Transformers
(OpenAI, 18)

True bidirectional?

Computationally fast

BERT
(Google, 18)

…

Roadmap Of Language Modeling

$$P(w_1, w_2, w_3, \dots, w_t) = ?$$

LM

Curse of dimensionality

N-gram

Distributional vectors

Can generalize to unseen sequences
"A cat is walking in the room."
"A dog is walking in the room."
"A dog is running in the kitchen."
…

SVD
(LSA: Boulder, 97)
(LDA: Berkeley, 03)

Feed-forward
neural net
(Bengio, 03)

Computationally slow

Word2vec
(Google,13)

CBOW

Skip gram

Global skip-gram

subword

GloVe
(Stanford, 14)

FastText
(FB, 16)

RNN/LSTM
(Mikolov, 10)

Contextualization

ELMo
(AI2&UW, 18)

Transformers
(OpenAI, 18)

True bidirectional?

Computationally fast

BERT
(Google, 18)

…

Roadmap Of Language Modeling

$P(w_1, w_2, w_3, \ldots, w_t) = ?$

LM

Curse of dimensionality

N-gram

Distributional vectors

SVD
(LSA: Boulder, 97)
(LDA: Berkeley, 03)

[A_{ij}]: word i appearing in doc j

Feed-forward neural net
(Bengio, 03)

Computationally slow

Word2vec
(Google,13)

CBOW

Skip gram

Global skip-gram

subword

GloVe
(Stanford, 14)

FastText
(FB, 16)

RNN/LSTM
(Mikolov, 10)

Contextualization

ELMo
(AI2&UW, 18)

Transformers
(OpenAI, 18)

True bidirectional?

Computationally fast

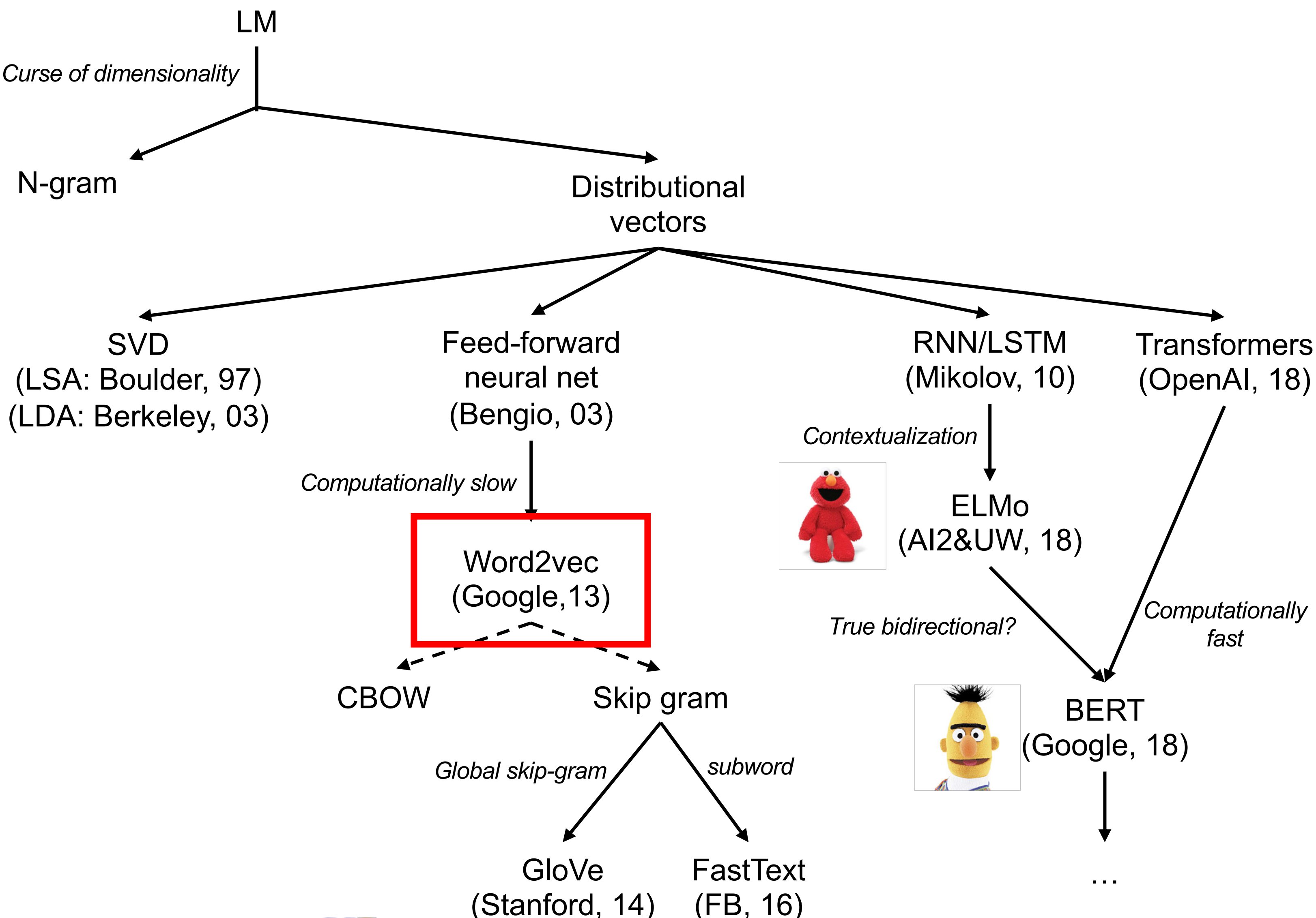
BERT
(Google, 18)

…

Cognitive Computation Group

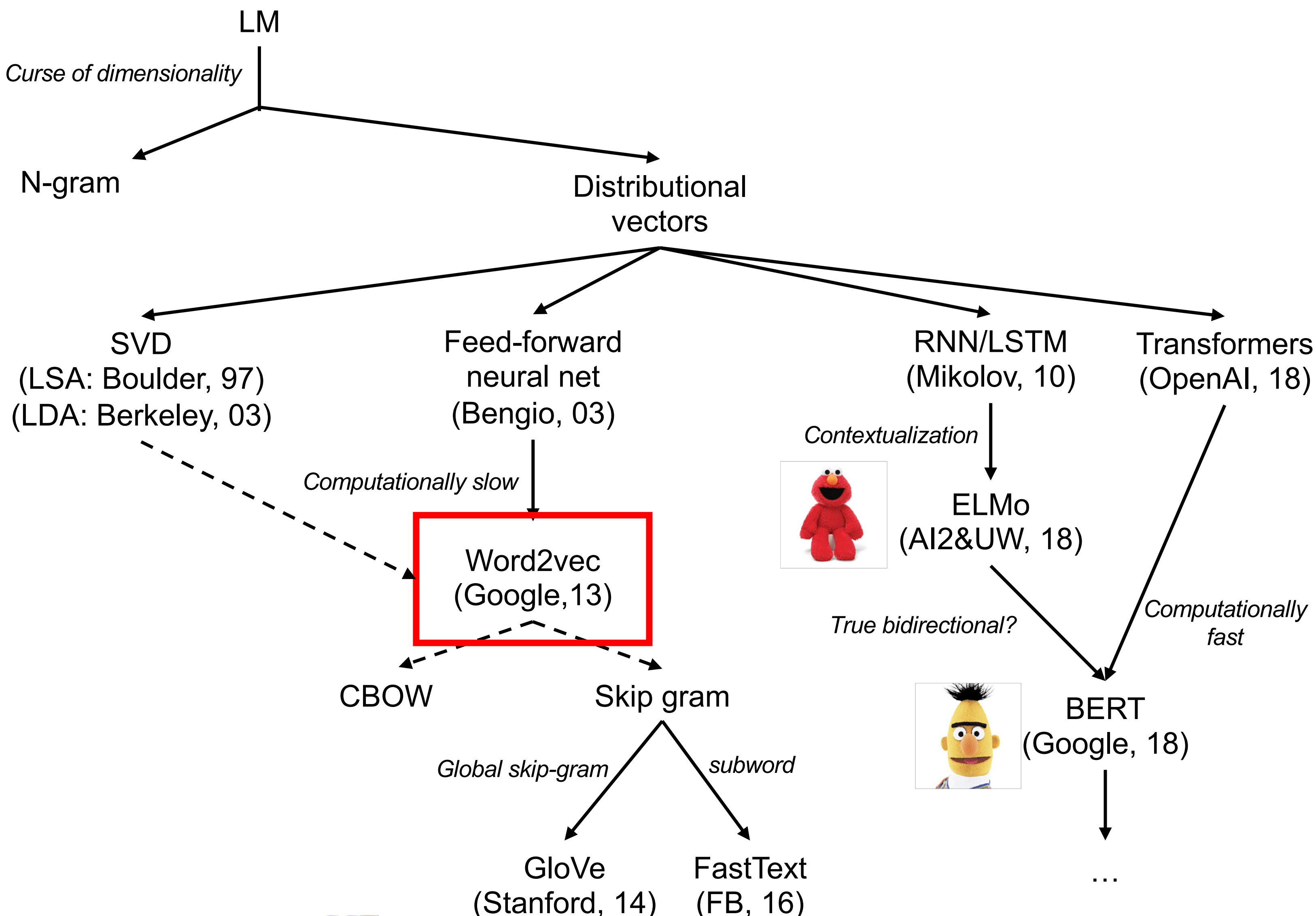
Roadmap Of Language Modeling

$$P(w_1, w_2, w_3, \ldots, w_t) = ?$$



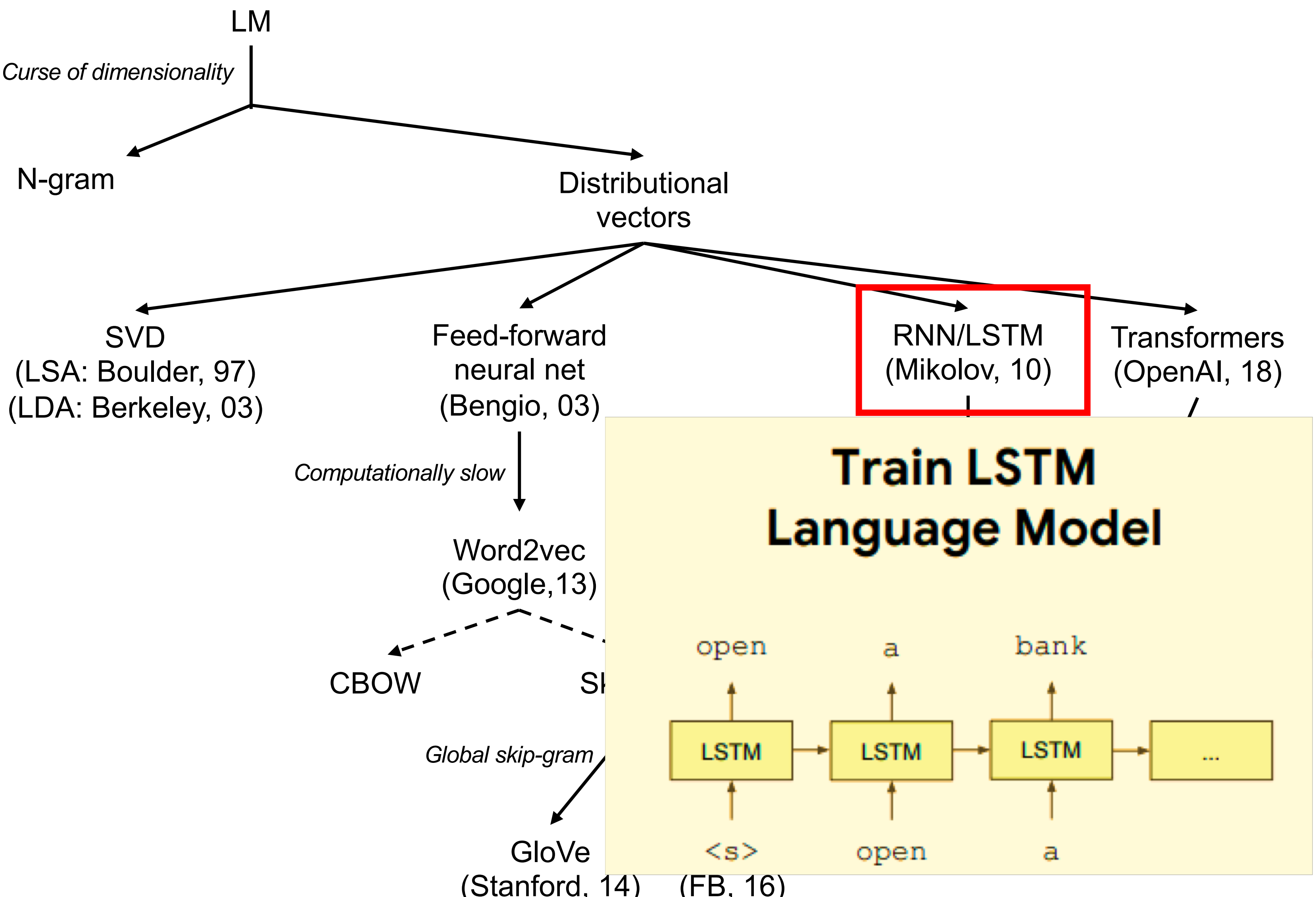
Roadmap Of Language Modeling

$P(w_1, w_2, w_3, \dots, w_t) = ?$



ROADMAP OF LANGUAGE MODELING

$$P(w_1, w_2, w_3, \dots, w_t) = ?$$



Roadmap Of Language Modeling

$P(w_1, w_2, w_3, \dots, w_t) = ?$

Problem: Word embeddings are applied in a context free manner

```
open a bank account          on the river bank

           [0.3, 0.2, -0.8, …]
```

SVD
(LSA: Boulder, 97)
(LDA: Berkeley, 03)

Feed-forward neural net
(Bengio, 03)

Computationally slow

Word2vec
(Google,13)

CBOW

Skip gram

Global skip-gram

subword

GloVe
(Stanford, 14)

FastText
(FB, 16)

RNN/LSTM
(Mikolov, 10)

Contextualization

ELMo
(AI2&UW, 18)

Transformers
(OpenAI, 18)

True bidirectional?

Computationally fast

BERT
(Google, 18)

…

Roadmap Of Language Modeling

$P(w_1, w_2, w_3, \ldots, w_t) = ?$

Problem: Word embeddings are applied in a context free manner

open a bank account   on the river bank

[0.3, 0.2, -0.8, …]

SVD
(LSA: Boulder, 97)
(LDA: Berkeley, 03)

Feed-forward
neural net
(Bengio, 03)

Computationally slow

Word2vec
(Google,13)

CBOW

Skip gram

Global skip-gram

subword

GloVe
(Stanford, 14)

FastText
(FB, 16)

RNN/LSTM
(Mikolov, 10)

Contextualization

ELMo
(AI2&UW, 18)

Transformers
(OpenAI, 18)

True bidirectional?

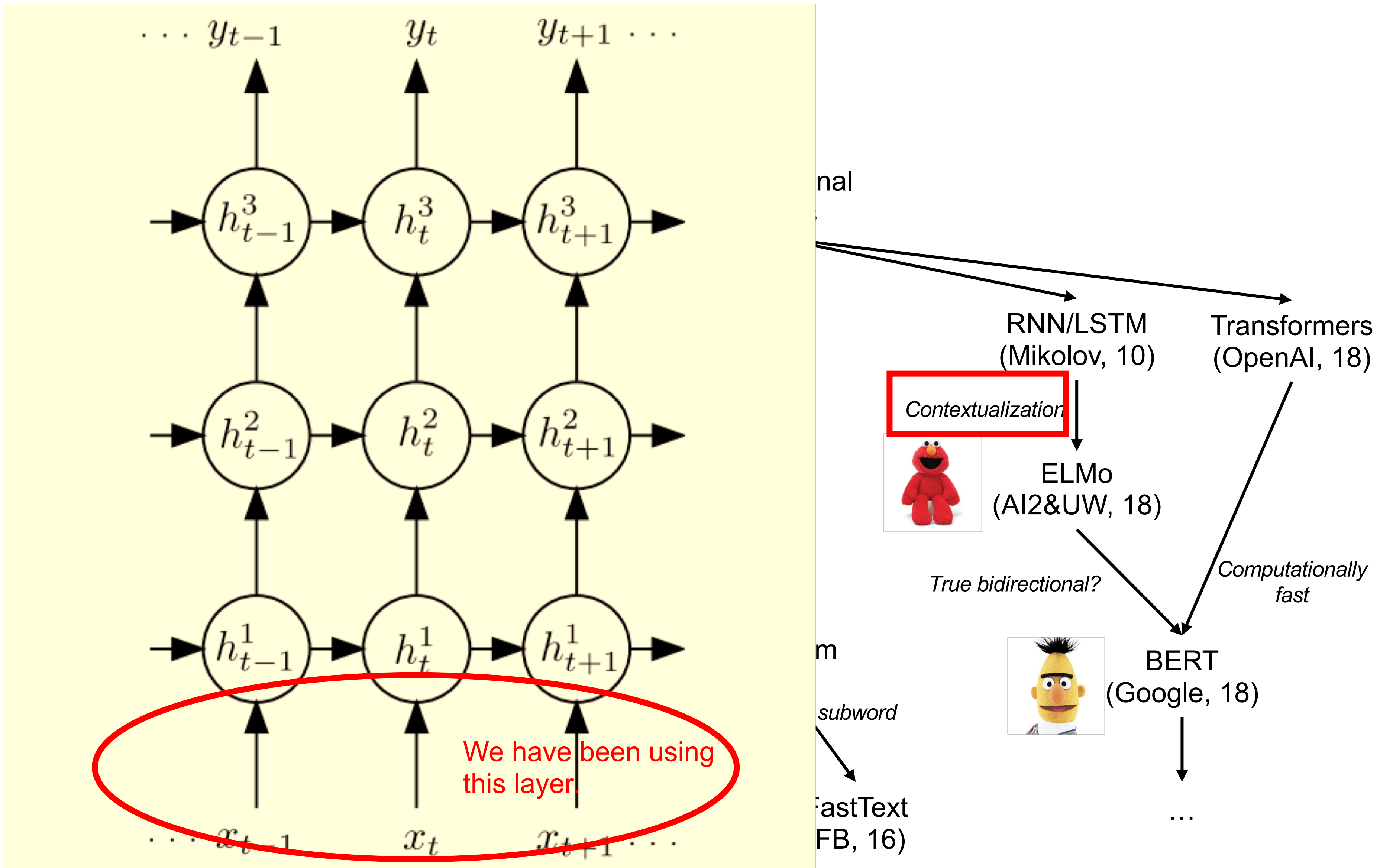
Computationally fast

BERT
(Google, 18)

…

Roadmap Of Language Modeling

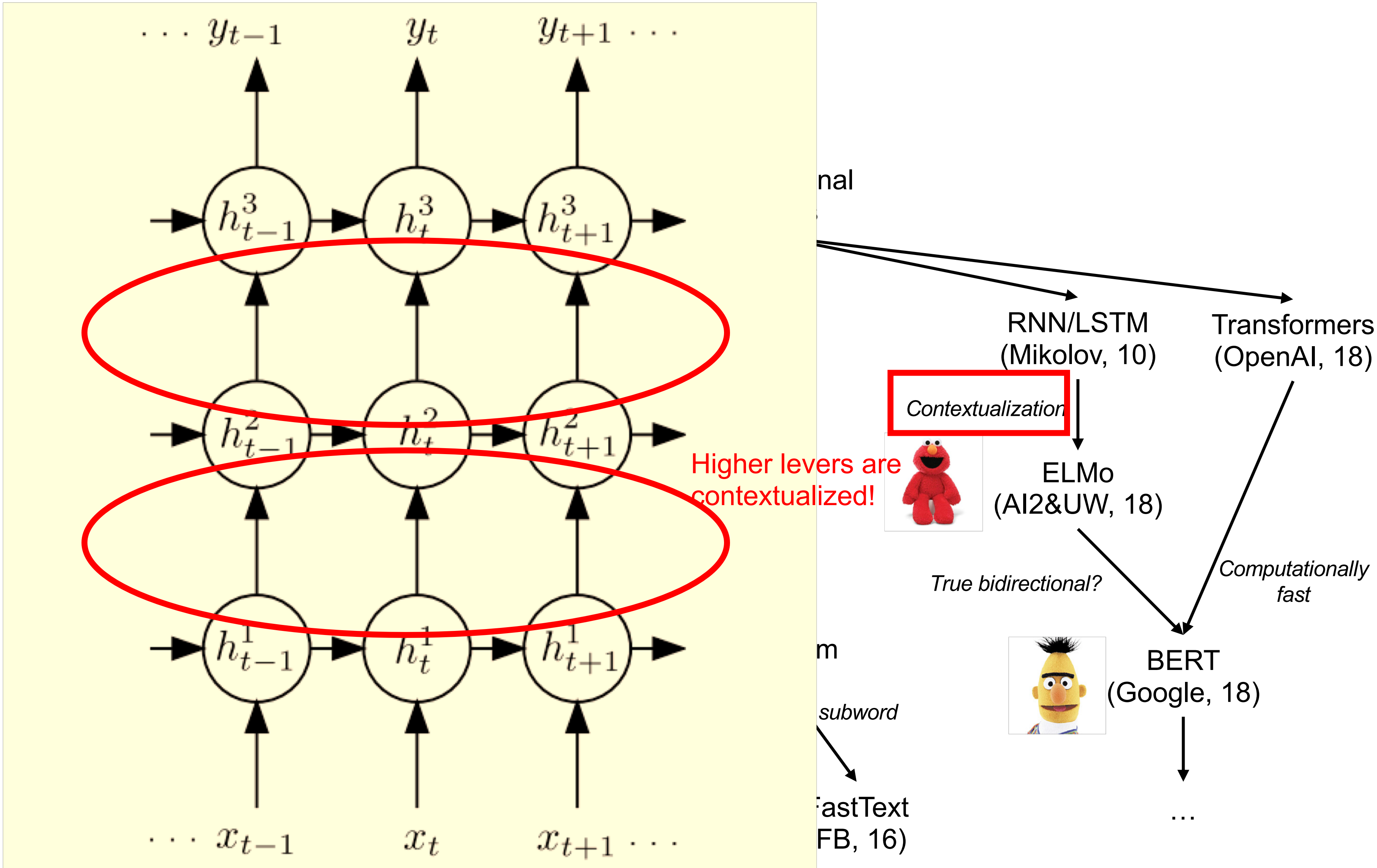
$$P(w_1, w_2, w_3, \dots, w_t) = ?$$



COGNITIVE COMPUTATION GROUP

Roadmap Of Language Modeling

$$P(w_1, w_2, w_3, \dots, w_t) = ?$$

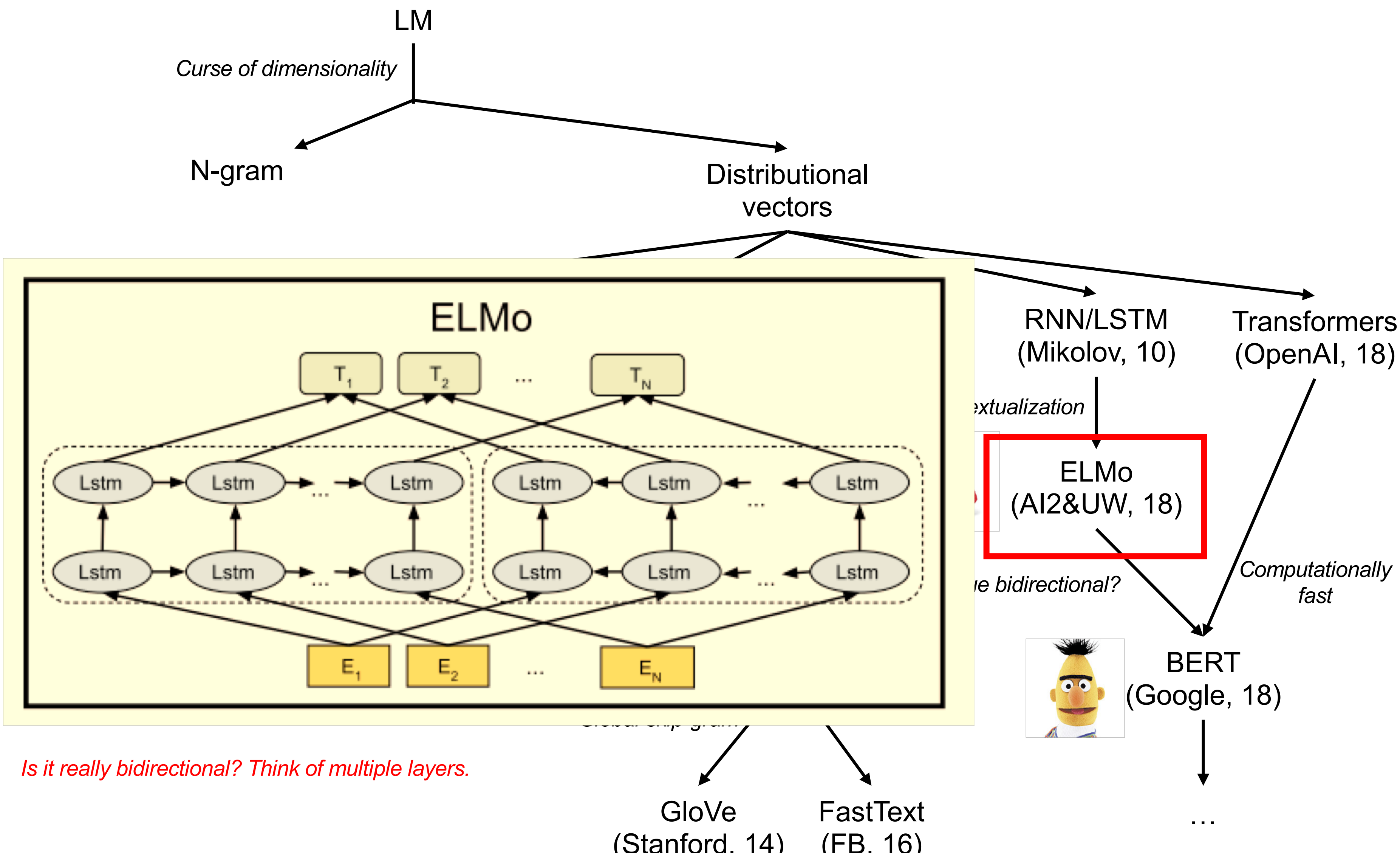


It Looks Naïve?

- Why did no one think of this before?
- Better question: Why wasn't contextual pre-training popular before 2018 with ELMo?
- Good results on pre-training is >1,000x to 100,000 more expensive than supervised training.
    - E.g., 10x-100x bigger model trained for 100x-1,000x as many steps.
    - Imagine it's 2013: Well-tuned 2-layer, 512-dim LSTM sentiment analysis gets 80% accuracy, training for 8 hours.
    - Pre-train LM on same architecture for a week, get 80.5%.
    - Conference reviewers: "Who would do something so expensive for such a small gain?"

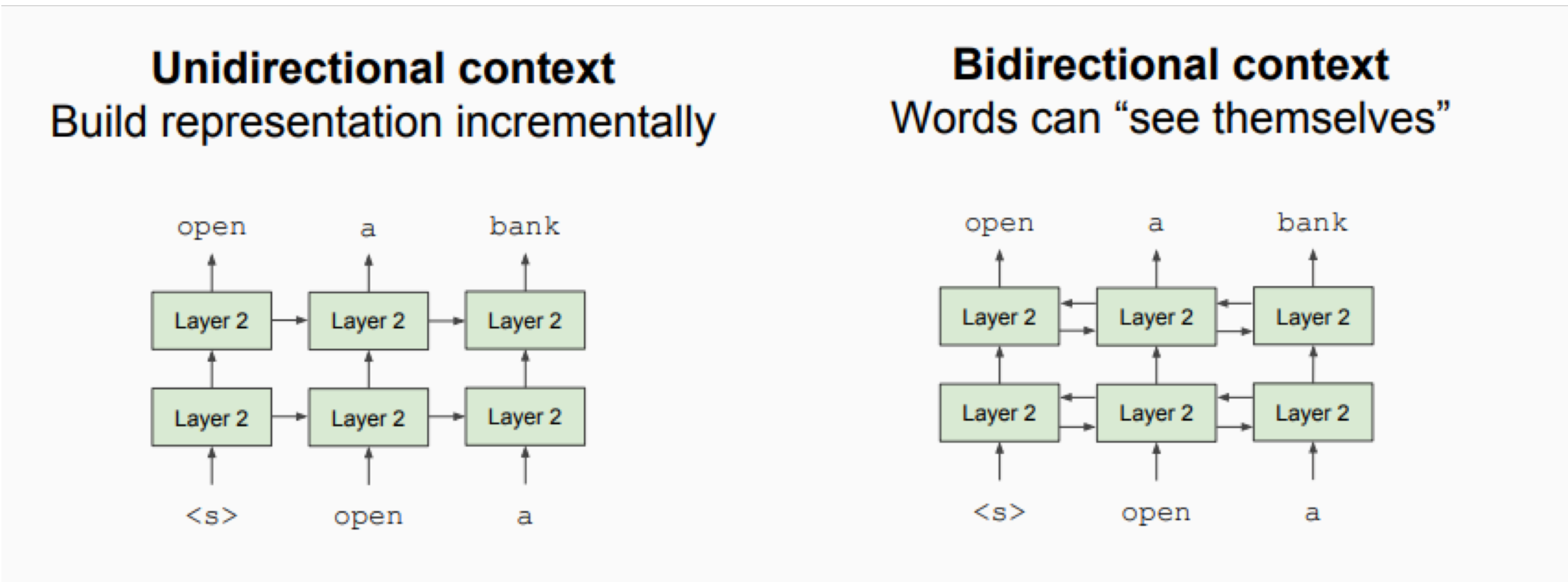
Roadmap Of Language Modeling

$P(w_1, w_2, w_3, \dots, w_t) = ?$

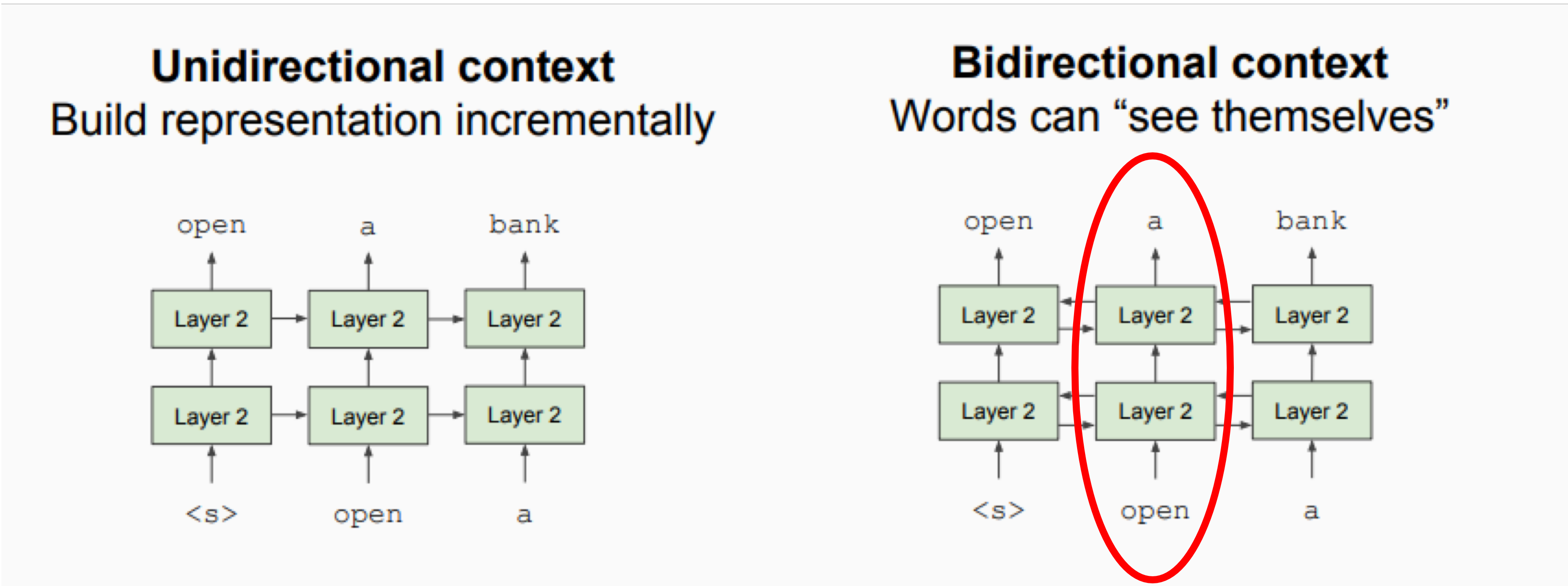


Is it really bidirectional? Think of multiple layers.

WHY CAN WE NOT DO TRUE BIDIRECTIONAL (USING LSTM)?



Why Can We Not Do True Bidirectional (Using LSTM)?



Assume we want to predict "a" when we see "open" [left to right]

But the layer above "open" has information about "a" from [right to left]

SOLUTION

- **Solution**: Mask out *k*% of the input words, and then predict the masked words
  - We always use $k = 15\%$

```
                              store                  gallon
                                ↑                       ↑
the man went to the [MASK] to buy a [MASK] of milk
```

- Too little masking: Too expensive to train
- Too much masking: Not enough context

In Addition To Mask LM: Next Sentence Prediction

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

To summarize what we have seen so far, BERT improves over previous methods by introducing "real" bidirectionality via mask LM, and on top of that, BERT further uses a multi-task learning setup to predict the relation between two adjacent sentences.

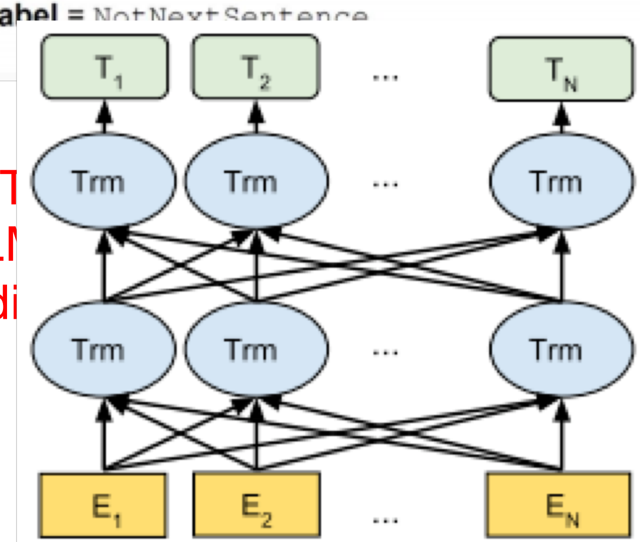
How is it implemented?--Transformers

IN ADDITION TO MASK LM: NEXT SENTENCE PREDICTION

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence



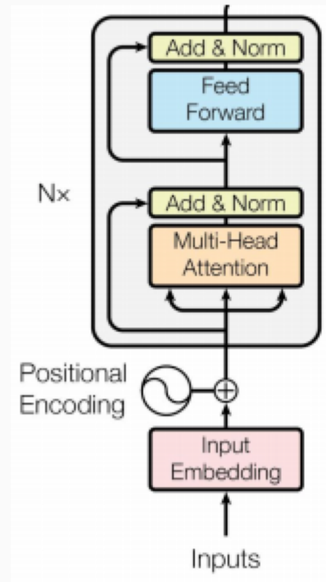
To summarize what we have seen so far, BERT ... ods
by introducing "real" bidirectionality via mask L...
further uses a multi-task learning setup to predi...
adjacent sentences.

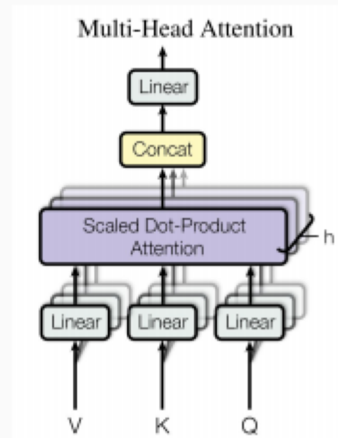
How is it implemented?--Transformers

TRANSFORMERS

Transformer encoder

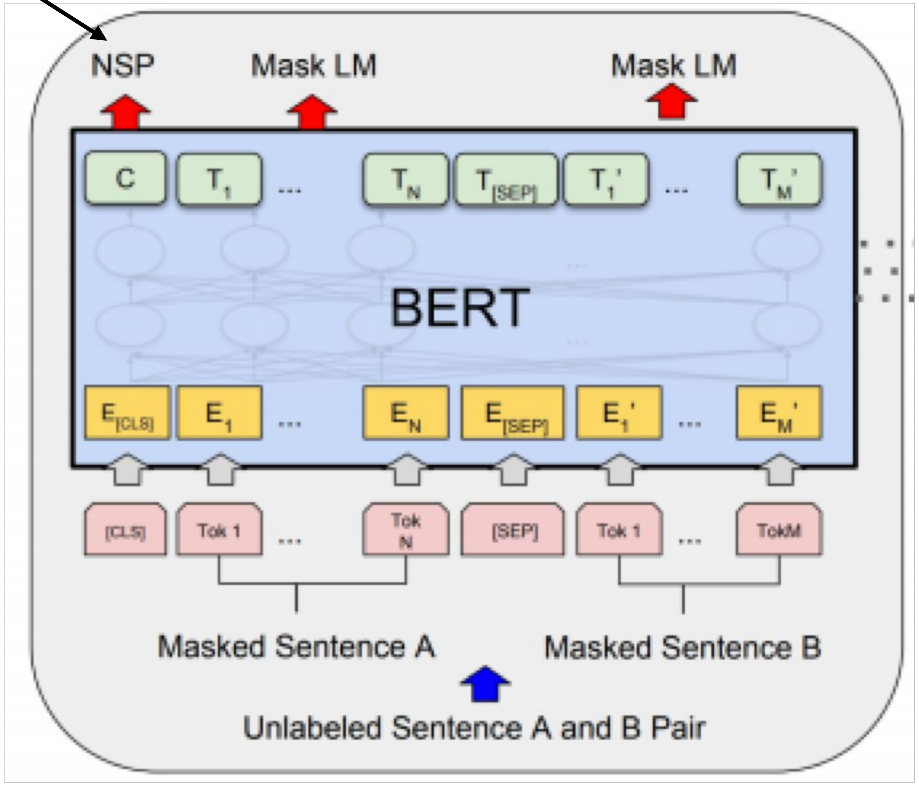
- Multi-headed self attention
  - Models context
- Feed-forward layers
  - Computes non-linear hierarchical features
- Layer norm and residuals
  - Makes training deep networks healthy
- Positional embeddings
  - Allows model to learn relative positioning





OVERVIEW

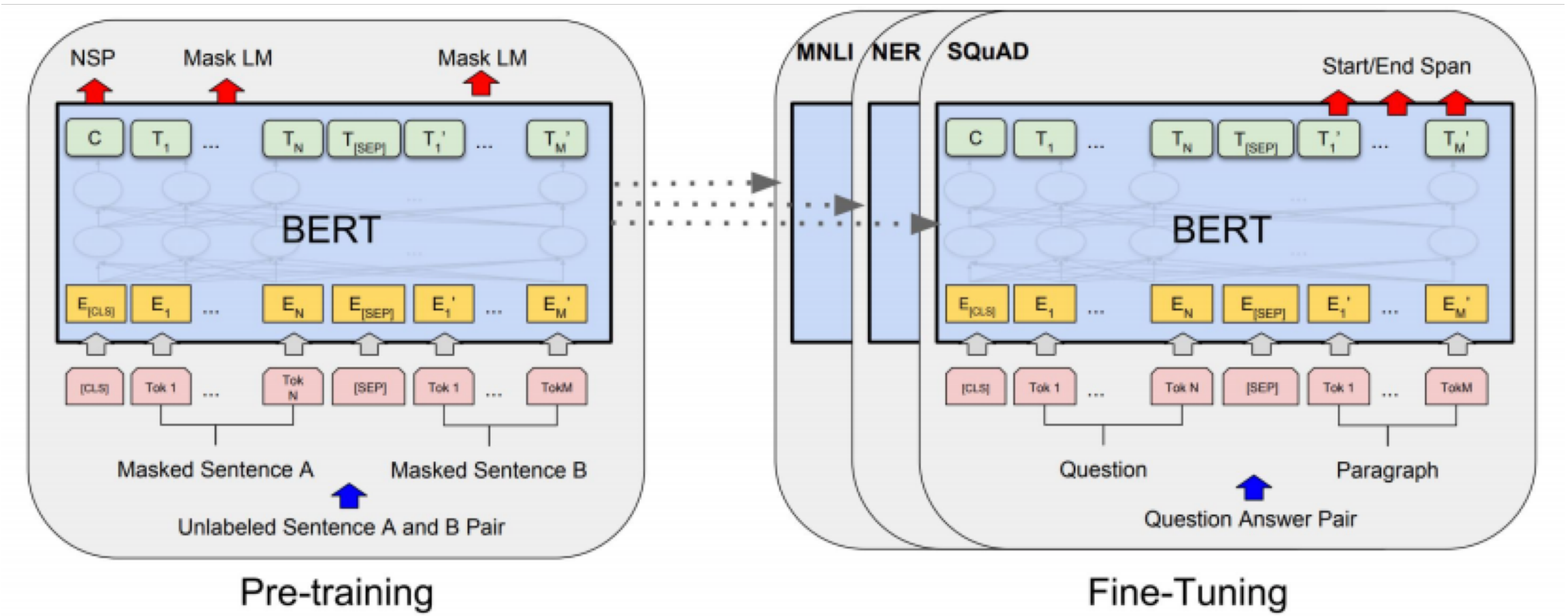
NSP: Next sentence prediction

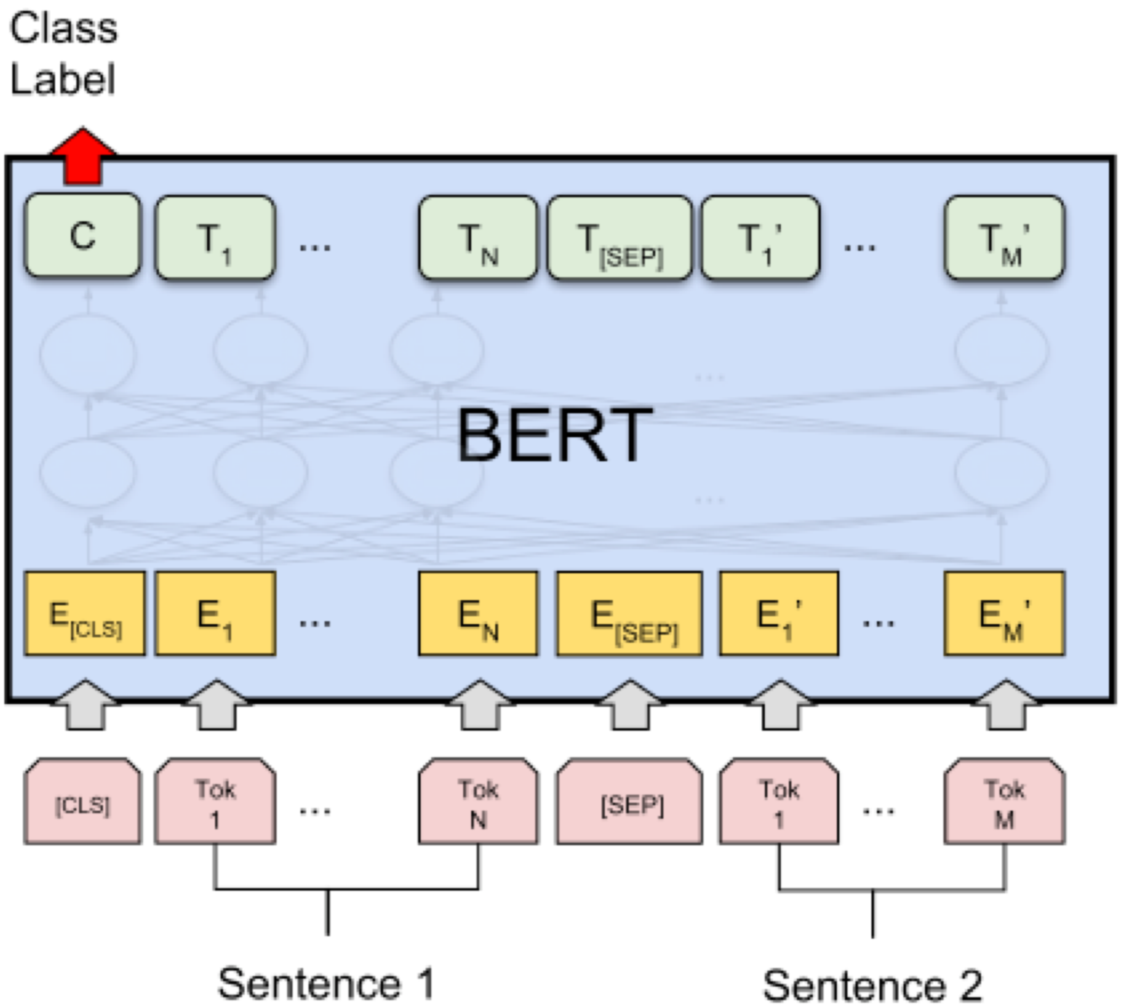


Training Details

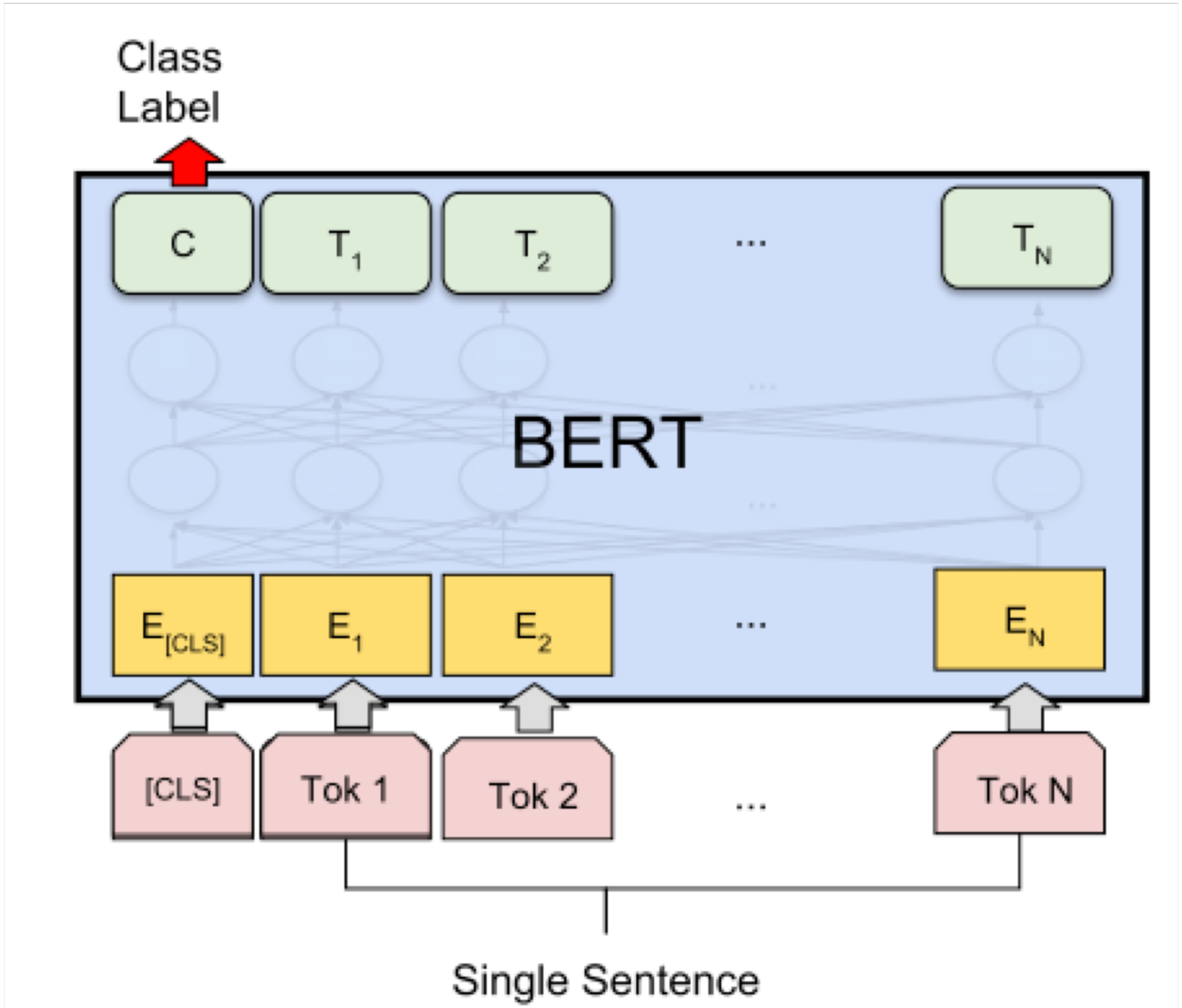
- <u>Data</u>: Wikipedia (2.5B words) + BookCorpus (800M words)
- <u>Batch Size</u>: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- <u>Training Time</u>: 1M steps (~40 epochs)
- <u>Optimizer</u>: AdamW, 1e-4 learning rate, linear decay
- `BERT-Base`: 12-layer, 768-hidden, 12-head
- `BERT-Large`: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

FINE-TUNING


NSP
Mask LM
Mask LM
C
BERT
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
BERT
Question
Paragraph
Question Answer Pair
Fine-Tuning



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

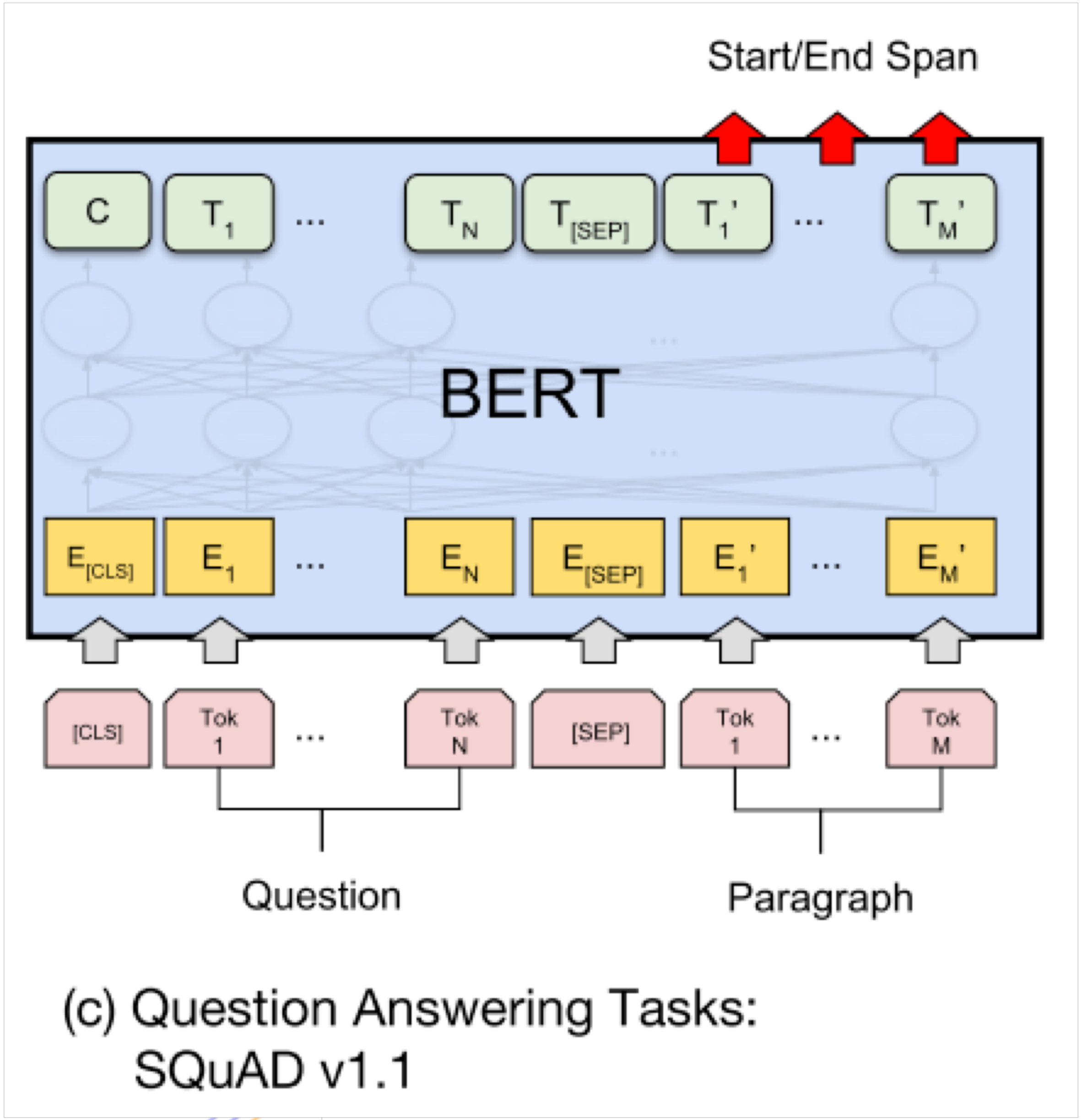


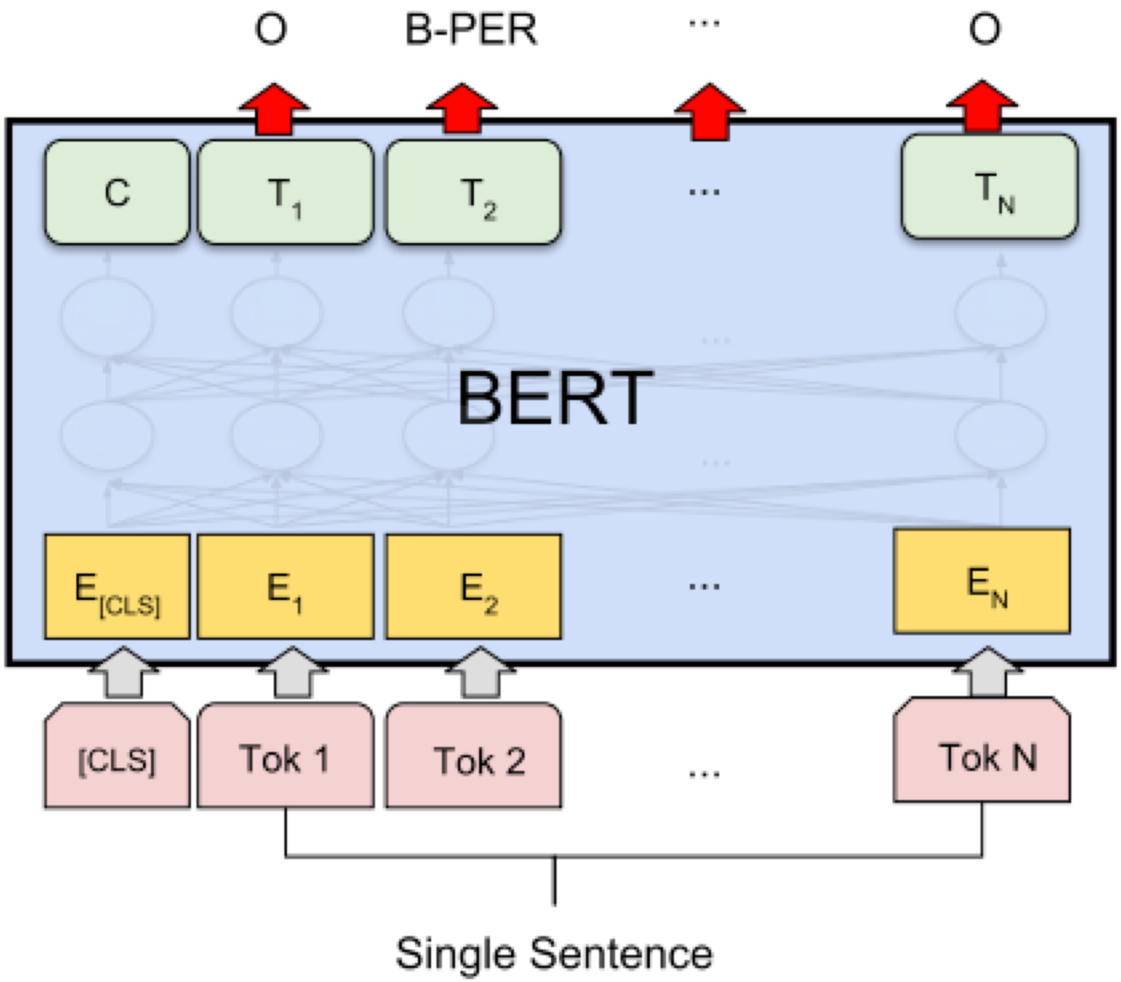
(b) Single Sentence Classification Tasks:
SST-2, CoLA

sentiment

Linguistically
acceptable

Learning a start and end vector from T_i





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

Tasks That Are Significantly Improved By BERT

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ = (L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

Tasks That Are Significantly Improved By BERT

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. Test results were scored against the hidden labels by the SWAG authors. †Human performance is measure with 100 samples, as reported in the SWAG paper.

Resources

- BERT [paper]: https://arxiv.org/abs/1810.04805
- BERT [presentation]: https://nlp.stanford.edu/seminar/details/jdevlin.pdf
- Transformers [blog]: http://jalammar.github.io/illustrated-transformer/
- Mask LM Demo By CogComp: http://orwell.seas.upenn.edu:4001/